

Lecture Notes in Artificial Intelligence 4203

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Floriana Esposito Zbigniew W. Raś
Donato Malerba Giovanni Semeraro (Eds.)

Foundations of Intelligent Systems

16th International Symposium, ISMIS 2006
Bari, Italy, September 27-29, 2006
Proceedings

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Floriana Esposito
Donato Malerba
Giovanni Semeraro
Università degli Studi di Bari
Dipartimento di Informatica
Via Orabona 4, 70126 Bari, Italy
E-mail: {esposito,malerba,semeraro}@di.uniba.it

Zbigniew W. Raś
University of North Carolina
Dept. of Computer Science
9201 University City Blvd., Charlotte, N.C. 28223, USA
E-mail: ras@uncc.edu

Library of Congress Control Number: 2006932794

CR Subject Classification (1998): I.2, H.3, H.2.8, H.4, H.5, F.1, I.5

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN	0302-9743
ISBN-10	3-540-45764-X Springer Berlin Heidelberg New York
ISBN-13	978-3-540-45764-0 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11875604 06/3142 5 4 3 2 1 0

Preface

*“Human knowledge and human power meet in one,
because where the cause is not known
the effect cannot be produced.”*

Francis Bacon

The 16th International Symposium on Methodologies for Intelligent Systems (ISMIS 2006) held in Bari, Italy, September 27–29, 2006, presented works performed by many scientists, worldwide on the development and exploitation of intelligent methods for solving complex problems. Starting from 1986, the ISMIS series has been held in Knoxville (USA), Charlotte (North Carolina), Turin (Italy), Trondheim (Norway), Warsaw (Poland), Zakopane (Poland), Lyon (France), Maebashi City (Japan) and Saratoga Springs (USA).

Among more than 200 submissions, the Program Committee selected, based on the opinion of at least two reviewers, 81 contributions, to be presented as long papers (if containing more complete work), or as short papers (if reporting about ongoing research). Many topics of interest to Artificial Intelligence research were covered: Active Media Human–Computer Interaction, Computational Intelligence, Intelligent Agent Technology, Intelligent Information Retrieval, Intelligent Information Systems, Knowledge Representation and Integration, Knowledge Discovery and Data Mining, Logic for AI and Logic Programming, Machine Learning, Text Mining and Web Intelligence. The symposium program also included three invited lectures, by Ivan Bratko, Ramon López de Mántaras and Steffen Staab.

Many people contributed in different ways to the success of the symposium and to this volume. The authors who continue to show their enthusiastic interest in intelligent systems research are a very important part of the success of scientific conferences. We highly appreciate the contribution of the Steering Committee, of the Program Committee members, and of all the additional reviewers that evaluated the submitted papers with efficiency and dedication. A special thanks goes to the members of the Organizing Committee, Nicola Di Mauro, Nicola Fanizzi and Stefano Ferilli, as well as to Teresa M.A. Basile, who worked hard at solving problems before and during the congress. We wish to thank the Dipartimento di Informatica of the University of Bari for encouraging and supporting us in organizing the event.

September 2006

Floriana Esposito
Zbigniew W. Raś
Donato Malerba
Giovanni Semeraro
Conference Chairs
ISMIS 2006

Organization

ISMIS 2006 was organized by the department of Computer Science, University of Bari.

Executive Committee

General Chair	Zbigniew W. Raś (UNC at Charlotte, USA)
Program Chair	Florian Esposito (Univ. of Bari, Italy)
Program Co-chairs	Donato Malerba (Univ. of Bari, Italy)
	Giovanni Semeraro (Univ. of Bari, Italy)
Organizing Committee	Nicola Di Mauro (Univ. of Bari, Italy)
	Nicola Fanizzi (Univ. of Bari, Italy)
	Stefano Ferilli (Univ. of Bari, Italy)

Steering Committee

Zbigniew W. Raś	(UNC at Charlotte, USA) Chair
Florian Esposito	(Univ. of Bari, Italy)
Mohand-Said Hacid	(Univ. Lyon 1, France)
David Hislop	(ARL/ARO, USA)
Robert Meersman	(Vrije Univ. Brussels, Belgium)
Neil Murray	(SUNY at Albany, USA)
Setsuo Ohsuga	(Waseda Univ., Japan)
Lorenza Saitta	(Univ. of Piemonte Orientale, Italy)
Shusaku Tsumoto	(Shimane Medical Univ., Japan)
Maria Zemankova	(NSF, USA)
Djamel Zighed	(Univ. Lyon 2, France)
Ning Zhong	(Maebashi Institute of Tech., Japan)

Program Committee

Luigia Carlucci Aiello, Italy	Sandra Carberry, USA
Aijun An, Canada	Juan Carlos Cubero, Spain
Troels Andreasen, Denmark	Agnieszka Dardzinska, Poland
Stefania Bandini, Italy	Ian Davidson, USA
Salima Benbernou, France	Robert Demolombe, France
Petr Berka, Czech Republic	Eric Dietrich, USA
Elisa Bertino, USA	Tapio Elomaa, Finland
Alan Biermann, USA	Attilio Giordana, Italy
Jacques Calmet, Germany	Marco Gori, Italy

Jerzy Grzymala-Busse, USA	Jan Rauch, Czech Republic
Mirsad Hadzikadic, USA	Gilbert Ritschard, Switzerland
Janusz Kacprzyk, Poland	Marie-Christine Rousset, France
Joost N. Kok, The Netherlands	Nahid Shahmehri, Sweden
Jacek Koronacki, Poland	Andrzej Skowron, Poland
T.Y. Lin, USA	Dominik Slezak, Canada
David Maluf, NASA Ames, USA	Steffen Staab, Germany
Ramon López de Mántaras, Spain	Olga Stepankova, Czech Republic
Simone Marinai, Italy	V.S. Subrahmanian, USA
Stan Matwin, Canada	Einoshin Suzuki, Japan
Paola Mello, Italy	Piero Torasso, Italy
Dunja Mladenić, Slovenia	Li-Shiang Tsay, USA
Hiroshi Motoda, Japan	Athena Vakali, Greece
Maria Teresa Pazienza, Italy	Juan Vargas, USA
Witold Pedrycz, Canada	Mario Vento, Italy
Luís Moniz Pereira, Portugal	Christel Vrain, France
James Peters, Canada	Alicja Wieczorkowska, Poland
Jean-Marc Petit, France	Xindong Wu, USA
Enric Plaza, Spain	Xintao Wu, USA
Vijay Raghavan, USA	Yiyu Yao, Canada

Referees

C. Åberg	K. Dellschaft	F. Riguzzi
J. Åberg	E. Diaz	C. Ringelstein
M. Aiello	C. Duma	C. Saathoff
A. Appice	A.M. Fanelli	A. Satyanarayana
R. Arndt	M. Gavanelli	D. Schall
N. Ashish	L. Guo	B. Schöler
P. Barahona	S. Guo	R. Singh
P. Basile	M. Kolta	S. Sizov
T.M.A. Basile	V. Koutsoukoulas	S. Storari
R. Basili	P. Kuntala	K. Stoupa
M. Berardi	P. Lambrix	L. Strömbäck
M. Biba	G. Lee	J. Takkinen
C. Caruso	O. Licchelli	H. Tan
M. Ceci	P. Lops	B. Tausch
G. Chen	G. Mao	P. Torroni
F. Chesani	N. Marques	G. Vizzari
C. d'Amato	G. Pallis	Y. Ye
A. De	I. Palmisano	V. Zou
M. Degemmis	S.S. Ravi	Y. Zhang
P. Dell'Acqua	D. Redavid	

Sponsoring Institutions

Dipartimento di Informatica, Università degli Studi di Bari

Table of Contents

Invited Talks

Lifecycle Knowledge Management: Getting the Semantics Across in X-Media	1
<i>Steffen Staab, Thomas Franz, Olaf Görlitz, Carsten Saathoff, Simon Schenk, Sergej Sizov</i>	
Argument-Based Machine Learning	11
<i>Ivan Bratko, Martin Možina, Jure Žabkar</i>	
Play It Again: A Case-Based Approach to Expressivity-Preserving Tempo Transformations in Music	18
<i>Ramon López de Mántaras</i>	

Active Media Human-Computer Interaction

Decision Fusion of Shape and Motion Information Based on Bayesian Framework for Moving Object Classification in Image Sequences	19
<i>Heungkyu Lee, JungHo Kim, June Kim</i>	
A Two-Stage Visual Turkish Sign Language Recognition System Based on Global and Local Features	29
<i>Hakan Haberdar, Songül Albayrak</i>	
Speech Emotion Recognition Using Spiking Neural Networks	38
<i>Cosimo A. Buscicchio, Przemysław Górecki, Laura Caponetti</i>	
Visualizing Transactional Data with Multiple Clusterings for Knowledge Discovery	47
<i>Nicolas Durand, Bruno Crémilleux, Einoshin Suzuki</i>	

Computational Intelligence

An Optimization Model for Visual Cryptography Schemes with Unexpanded Shares	58
<i>Ching-Sheng Hsu, Shu-Fen Tu, Young-Chang Hou</i>	
A Fast Temporal Texture Synthesis Algorithm Using Segment Genetic Algorithm	68
<i>Wen-hui Li, Yu Meng, Zhen-hua Zhang, Dong-fei Liu, Jian-yuan Wang</i>	

Quantum-Behaved Particle Swarm Optimization with Immune Operator	77
<i>Jing Liu, Jun Sun, Wenbo Xu</i>	
Particle Swarm Optimization-Based SVM for Incipient Fault Classification of Power Transformers.....	84
<i>Tsair-Fwu Lee, Ming-Yuan Cho, Chin-Shiuh Shieh, Hong-Jen Lee, Fu-Min Fang</i>	
AntTrend: Stigmergetic Discovery of Spatial Trends	91
<i>Ashkan Zarnani, Masoud Rahgozar, Caro Lucas, Azizollah Memariani</i>	
Genetic Algorithm Based Approach for Multi-UAV Cooperative Reconnaissance Mission Planning Problem	101
<i>Jing Tian, Lincheng Shen, Yanxing Zheng</i>	
Improving SVM Training by Means of NTIL When the Data Sets Are Imbalanced	111
<i>Carlos E. Vivaracho</i>	
Evolutionary Induction of Cost-Sensitive Decision Trees.....	121
<i>Marek Krętowski, Marek Grześ</i>	
Triangulation of Bayesian Networks Using an Adaptive Genetic Algorithm	127
<i>Hao Wang, Kui Yu, Xindong Wu, Hongliang Yao</i>	
Intelligent Agent Technology	
Intelligent Agents That Make Informed Decisions	137
<i>John Debenham, Elaine Lawrence</i>	
Using Intelligent Agents in e-Government for Supporting Decision Making About Service Proposals	147
<i>Pasquale De Meo, Giovanni Quattrone, Domenico Ursino</i>	
A Butler Agent for Personalized House Control	157
<i>Berardina De Carolis, Giovanni Cozzolongo, Sebastiano Pizzutilo</i>	
Incremental Aggregation on Multiple Continuous Queries	167
<i>Chun Jin, Jaime Carbonell</i>	

Location-Aware Multi-agent Based Intelligent Services in Home Networks	178
<i>Minsoo Lee, Yong Kim, Yoonsik Uhm, Zion Hwang, Gwanyeon Kim, Sehyun Park, Ohyoung Song</i>	
A Verifiable Logic-Based Agent Architecture	188
<i>Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello</i>	

Intelligent Information Retrieval

Flexible Querying of XML Documents	198
<i>Krishnaprasad Thirunarayan, Trivikram Immaneni</i>	
VIRMA: Visual Image Retrieval by Shape MAtching	208
<i>Giovanna Castellano, Ciro Castiello, Anna Maria Fanelli</i>	
Asymmetric Page Split Generalized Index Search Trees for Formal Concept Analysis	218
<i>Ben Martin, Peter Eklund</i>	
Blind Signal Separation of Similar Pitches and Instruments in a Noisy Polyphonic Domain	228
<i>Rory A. Lewis, Xin Zhang, Zbigniew W. Ras</i>	
Score Distribution Approach to Automatic Kernel Selection for Image Retrieval Systems	238
<i>Anca Doloc-Mihu, Vijay V. Raghavan</i>	
Business Intelligence in Large Organizations: Integrating Which Data?	248
<i>David Maluf, David Bell, Naveen Ashish, Peter Putz, Yuri Gawdiak</i>	

Intelligent Infomation Systems

Intelligent Methodologies for Scientific Conference Management	258
<i>Marenglen Biba, Stefano Ferilli, Nicola Di Mauro, Teresa Maria Altomare Basile</i>	
The Consistent Data Replication Service for Distributed Computing Environments	268
<i>Jaechun No, Chang Won Park, Sung Soon Park</i>	

OntoBayes Approach to Corporate Knowledge	274
<i>Yi Yang, Jacques Calmet</i>	
About Inclusion-Based Generalized Yes/No Queries in a Possibilistic Database Context	284
<i>Patrick Bosc, Nadia Lietard, Olivier Pivert</i>	
Flexible Querying of an Intelligent Information System for EU Joint Project Proposals in a Specific Topic	290
<i>Stefan Trausan-Matu, Ruxandra Bantea, Vlad Posea, Diana Petrescu, Alexandru Gartner</i>	
Multi-expression Face Recognition Using Neural Networks and Feature Approximation	296
<i>Adnan Khashman, Akram Abu Garad</i>	
A Methodology for Building Semantic Web Mining Systems	306
<i>Francesca A. Lisi</i>	
Content-Based Image Filtering for Recommendation	312
<i>Kyung-Yong Jung</i>	

Knowledge Representation and Integration

A Declarative Kernel for $\mathcal{ALC}$ Concept Descriptions	322
<i>Nicola Fanizzi, Claudia d'Amato</i>	
RDF as Graph-Based, Diagrammatic Logic	332
<i>Frithjof Dau</i>	
A Framework for Defining and Verifying Clinical Guidelines: A Case Study on Cancer Screening	338
<i>Federico Chesani, Pietro De Matteis, Paola Mello, Marco Montali, Sergio Storari</i>	
Preferred Generalized Answers for Inconsistent Databases	344
<i>Luciano Caroprese, Sergio Greco, Irina Trubitsyna, Ester Zumpano</i>	
Representation Interest Point Using Empirical Mode Decomposition and Independent Components Analysis	350
<i>Dongfeng Han, Wenhui Li, Xiaosuo Lu, Yi Wang, Ming Li</i>	
Integration of Graph Based Authorization Policies	359
<i>Chun Ruan, Vijay Varadharajan</i>	

Knowledge Discovery and Data Mining

Supporting Visual Exploration of Discovered Association Rules Through Multi-Dimensional Scaling	369
<i>Margherita Berardi, Annalisa Appice, Corrado Loglisci, Pietro Leo</i>	
Evaluating Learning Algorithms for a Rule Evaluation Support Method Based on Objective Rule Evaluation Indices	379
<i>Hidenao Abe, Shusaku Tsumoto, Miho Ohsaki, Takahira Yamaguchi</i>	
Quality Assessment of k -NN Multi-label Classification for Music Data	389
<i>Alicja Wieczorkowska, Piotr Synak</i>	
Effective Mining of Fuzzy Multi-Cross-Level Weighted Association Rules	399
<i>Mehmet Kaya, Reda Alhajj</i>	
A Methodological Contribution to Music Sequences Analysis	409
<i>Daniele P. Radicioni, Marco Botta</i>	
Towards a Framework for Inductive Querying	419
<i>Jeroen S. de Bruin</i>	
Towards Constrained Co-clustering in Ordered 0/1 Data Sets	425
<i>Ruggero G. Pensa, Céline Robardet, Jean-François Boulicaut</i>	
A Comparative Analysis of Clustering Methodology and Application for Market Segmentation: K-Means, SOM and a Two-Level SOM	435
<i>Sang-Chul Lee, Ja-Chul Gu, Yung-Ho Suh</i>	
Action Rules Discovery, a New Simplified Strategy	445
<i>Zbigniew W. Raś, Agnieszka Dardzińska</i>	
Characteristics of Indiscernibility Degree in Rough Clustering Examined Using Perfect Initial Equivalence Relations	454
<i>Shoji Hirano, Shusaku Tsumoto</i>	
Implication Strength of Classification Rules	463
<i>Gilbert Ritschard, Djamel A. Zighed</i>	
A New Clustering Approach for Symbolic Data and Its Validation: Application to the Healthcare Data	473
<i>Haytham Elghazel, Véronique Deslandres, Mohand-Said Hacid, Alain Dussauchoy, Hamamache Kheddouci</i>	

Action Rules Discovery System DEAR ₃	483
<i>Li-Shiang Tsay, Zbigniew W. Raś</i>	
Mining and Modeling Database User Access Patterns	493
<i>Qingsong Yao, Aijun An, Xiangji Huang</i>	

Logic for AI and Logic Programming

Belief Revision in the Situation Calculus Without Plausibility Levels	504
<i>Robert Demolombe, Pilar Pozos Parra</i>	
Norms, Institutional Power and Roles: Towards a Logical Framework	514
<i>Robert Demolombe, Vincent Louis</i>	
Adding Efficient Data Management to Logic Programming Systems	524
<i>Giorgio Terracina, Nicola Leone, Vincenzino Lio, Claudio Panetta</i>	
A Logic-Based Approach to Model Supervisory Control Systems	534
<i>Pierangelo Dell'Acqua, Anna Lombardi, Luís Moniz Pereira</i>	
SAT as an Effective Solving Technology for Constraint Problems	540
<i>Marco Cadoli, Toni Mancini, Fabio Patrizi</i>	
Dependency Tree Semantics	550
<i>Leonardo Lesmo, Livio Robaldo</i>	

Machine Learning

Mining Tolerance Regions with Model Trees	560
<i>Annalisa Appice, Michelangelo Ceci</i>	
Lazy Learning from Terminological Knowledge Bases	570
<i>Claudia d'Amato, Nicola Fanizzi</i>	
Diagnosis of Incipient Fault of Power Transformers Using SVM with Clonal Selection Algorithms Optimization	580
<i>Tsair-Fwu Lee, Ming-Yuan Cho, Chin-Shiuh Shieh, Hong-Jen Lee, Fu-Min Fang</i>	
An Overview of Alternative Rule Evaluation Criteria and Their Use in Separate-and-Conquer Classifiers	591
<i>Fernando Berzal, Juan-Carlos Cubero, Nicolás Marín, José-Luis Polo</i>	

Learning Students' Learning Patterns with Support Vector Machines	601
<i>Chao-Lin Liu</i>	
Practical Approximation of Optimal Multivariate Discretization	612
<i>Tapio Elomaa, Jussi Kujala, Juho Rousu</i>	
Optimisation and Evaluation of Random Forests for Imbalanced Datasets	622
<i>Julien Thomas, Pierre-Emmanuel Jouve, Nicolas Nicoloyannis</i>	
Improving SVM-Linear Predictions Using CART for Example Selection	632
<i>João M. Moreira, Alípio M. Jorge, Carlos Soares, Jorge Freire de Sousa</i>	
Simulated Annealing Algorithm with Biased Neighborhood Distribution for Training Profile Models	642
<i>Anton Bezuglov, Juan E. Vargas</i>	
A Conditional Model for Tonal Analysis	652
<i>Daniele P. Radicioni, Roberto Esposito</i>	
Hypothesis Diversity in Ensemble Classification	662
<i>Lorenza Saitta</i>	
Complex Adaptive Systems: Using a Free-Market Simulation to Estimate Attribute Relevance	671
<i>Christopher N. Eichelberger, Mirsad Hadžikadić</i>	
Text Mining	
Exploring Phrase-Based Classification of Judicial Documents for Criminal Charges in Chinese	681
<i>Chao-Lin Liu, Chwen-Dar Hsieh</i>	
Regularization for Unsupervised Classification on Taxonomies	691
<i>Diego Sona, Sriharsha Veeramachaneni, Nicola Polettini, Paolo Avesani</i>	
A Proximity Measure and a Clustering Method for Concept Extraction in an Ontology Building Perspective	697
<i>Guillaume Cleuziou, Sylvie Billot, Stanislas Lew, Lionel Martin, Christel Vrain</i>	

An Intelligent Personalized Service for Conference Participants	707
<i>Marco Degemmis, Pasquale Lops, Pierpaolo Basile</i>	
Contextual Maps for Browsing Huge Document Collections	713
<i>Krzysztof Ciesielski, Mieczysław A. Kłopotek</i>	

Web Intelligence

Classification of Polish Email Messages: Experiments with Various Data Representations	723
<i>Jerzy Stefanowski, Marcin Zienkiewicz</i>	
Combining Multiple Email Filters Based on Multivariate Statistical Analysis	729
<i>Wenbin Li, Ning Zhong, Chunnian Liu</i>	
Employee Profiling in the Total Reward Management	739
<i>Silverio Petruzzellis, Oriana Licchelli, Ignazio Palmisano, Valeria Bavaro, Cosimo Palmisano</i>	
Mining Association Rules in Temporal Document Collections	745
<i>Kjetil Nørvåg, Trond Øivind Eriksen, Kjell-Inge Skogstad</i>	
Self-supervised Relation Extraction from the Web	755
<i>Ronen Feldman, Benjamin Rosenfeld, Stephen Soderland, Oren Etzioni</i>	
Author Index	765

Lifecycle Knowledge Management: Getting the Semantics Across in X-Media

Steffen Staab, Thomas Franz, Olaf Görlitz, Carsten Saathoff, Simon Schenk,
and Sergej Sizov

ISWeb, University of Koblenz-Landau, Germany
{staab, franz, goerlitz, saathoff, sschenk, sizov}@uni-koblenz.de
<http://isweb.uni-koblenz.de/>

Abstract. Knowledge and information spanning multiple information sources, multiple media, multiple versions and multiple communities challenge the capabilities of existing knowledge and information management infrastructures by far — primarily in terms of intellectually exploiting the stored knowledge and information. In this paper we present some semantic web technologies of the EU integrated project X-Media that build bridges between the various information sources, the different media, the stations of knowledge management and the different communities. Core to this endeavour is the combination of information extraction with formal ontologies as well as with semantically lightweight folksonomies.

1 Management of Lifecycle Knowledge

Knowledge and information management faces larger challenges than ever, not in spite, but because of their successes. Spanning *multiple information sources*, *multiple media*, *multiple versions* and *multiple communities* challenges the capabilities of existing information management infrastructures by far — not primarily in terms of storage size and network bandwidth, but in terms of intellectually exploiting the stored knowledge. The rationale behind the growth of available resources lies in the possibility of digital recording and the objective of optimizing business processes wherever possible.

In the EU IST project “X-Media — Large scale knowledge sharing and reuse across media”¹ we approach these challenges for management of product lifecycle knowledge for complex industrial products (such as cars or jet engines) by targeting the

- Understanding of content from across multiple media formats and resources;
- Representation of dynamic and uncertain knowledge including provenance, change and process knowledge;

¹ This work was funded by the X-Media project (www.x-media-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978. The view of the authors does not necessarily represent the view of the X-Media project as a whole.

- Semantic user interaction tightly embedding paradigms of searching, retrieval, browsing, access and re-use into a knowledge sharing infrastructure.

We target the management of lifecycle knowledge allowing for run-through knowledge extraction, capturing, representation, and re-use. In this paper we elaborate the scenario of problem analysis and resolution that we approach at the industrial use-cases partners of our project, Rolls Royce and Fiat. We highlight the semantic infrastructure technology that we estimate will help us to achieve our goals of product lifecycle management.

2 Scenario

The shortcomings of existing knowledge management systems become clear in situations that require complex analysis of the product lifecycle, such as resolution of non-trivial technical problems. What happens, when maintenance technicians or pilots perceive unusual sound or vibration of a jet engine? Usually, the relevant pieces of problem-related information are scattered across several systems, data repositories, and media types. Consequently, the systematic problem analysis requires expert knowledge and experience in different areas of the product lifecycle. The lack of flexible mechanisms for finding problems with similar provenance and identifying ways to resolve them in the past would force the manufacturer to ask through a number of experts or departments. It is not surprising that the resulting necessary consultations may delay the actual problem resolution by several weeks or months. What could improve the efficiency of this time-consuming procedure?

2.1 User Requirements

The complex problem analysis typically requires a flexible and integrated view on all aspects of the product lifecycle. Furthermore, the framework should allow the tight cooperation between manufacturer, product suppliers, and customers for flexible representation and sharing of their requirements and demands. This objective leads to following major user requirements.

Integrated workflows. The lifecycle of complex systems is documented in their individual history including maintenance reports, fault descriptions, records on operational conditions, etc. Every time the product (e.g. jet engine) is serviced or inspected, financial information is generated. If problems are found, pictures are taken, reports are written. It is clear that knowledge acquisition should be seamlessly integrated into all workflows of the product lifecycle.

Tracking of provenance. During lifecycle monitoring, much secondary knowledge is generated and exchanged regarding the sources of information, backgrounds and motivation of decisions made, or responsibilities for these decisions (e.g. what type of sensors was used, which maintenance operations preceded the anomaly, or why the technician decided to suspend the seemingly health engine).

This provenance knowledge can be used for fault isolation, recalling common activities for frequent maintenance scenarios, framework accounting, or workflow optimization purposes.

Knowledge sharing. It could be expected that the continuously increasing amount of available information during the product lifecycle facilitates the product design and solving of observed problems. However, meaningful interpretation of this information typically requires a big portion of background knowledge, experience, and/or expert terminology that are not represented by the corresponding system at all. The knowledge exploitation framework should provide flexible mechanisms for knowledge sharing and interpretation.

Ease of cross-media problem analysis. Existing systems typically provide an isolated view on one type of media (e.g. sensor data, textual explanations, X-ray, thermal paint, or photography digital images). Each media type comes with domain-specific attributes and features that are crucial for understanding in the context of analyzed problem, such as recognition of deformations or surface erosion. Since particular issue attachments (email chains, textual problem reports, pictures of damaged parts) are usually stored separately, the cross-media analysis becomes the bottleneck of the problem resolution. The framework should provide systematic view on all relevant problem facets across different media types and data formats.

2.2 Technical Requirements

Scalability. The broad use of sensor networks allows fine-grained collection of physical parameters (temperature, sound, vibration, etc.) that characterize the operation of complex systems. The amount of data produced by modern monitoring tools is rapidly increasing. The records for each individual system describing its whole lifecycle can easily sum up to several Terabytes of information. An important requirement is therefore the high efficiency and scalability of interactive search/retrieval algorithms and expert tools.

Flexible representation. Monitoring physical properties is always incident with measurement errors that can be captured by averaging, smoothing, or providing tolerance/confidence intervals. Many sources of human knowledge, such as maintenance plans, user observations (e.g. pilot reports), or expert recommendations (e.g. problem resolution protocols), show some grade of uncertainty or are partially contradictory. Ideally, the framework should be able to cope with incomplete and uncertain knowledge and assist the expert user to navigate through the information space in a focused manner.

Infrastructure. An important aspect of the knowledge management framework is clearly the flexible architecture for sharing and re-use of expert knowledge. The framework should support collaborative knowledge organization beyond the borders of particular systems, expert groups, application scenarios, or media

types. To this reason, it should offer to each expert user the full autonomy of information management and freedom of data organization. On the other hand, the framework should provide mechanisms for easy and instant knowledge sharing, e.g. by annotating all relevant pieces of the problem report with the name of the responsible department, and associating group access with this annotation.

2.3 Focus of This Paper

It is clear that there is no simple solution to the introduced versatile complex of requirements. In this proposal, we restrict ourselves to the following major facets of the desired knowledge management framework:

- seamless *integration of knowledge* from different sources
- *flexible infrastructure* for knowledge representation for various application-driven contexts
- infrastructure for easy and instant *knowledge sharing* between experts, organizations, or platforms
- representation of expert *background knowledge* for better problem understanding
- *tracking of provenance* for decisions, changes, and data items
- ease of *knowledge acquisition* that should not require substantial additional efforts

On the one hand, our vision aims to overcome traditional limitations of data-driven information systems and addresses a wide range of applications with non-trivial requirements, such as product lifecycle management. On the other hand, the resulting technology forms the basis of the framework for knowledge sharing and re-use and provides direct support for its further components (such as information extraction or security mechanisms).

3 Design Decisions

Following we present the design decisions made to cover the requirements of Section 2.3. Table 1 summarizes the stated requirements and indicates, which core technology addresses a specific requirement. Further, we explain the technologies employed, how they help us in achieving our goal and how they integrate with each other.

Table 1. Requirements and technologies to cover them

Requirements \ Design Decisions	Tagging	Ontologies	Semantic Desktop	P2P	RDF
<i>integration of knowledge</i>	×	×	×		×
<i>knowledge sharing</i>	×			×	
<i>flexible infrastructure</i>				×	×
<i>knowledge acquisition</i>	×	×	×		
<i>tracking of provenance</i>			×		×
<i>background knowledge</i>		×			

Tagging. Tagging denotes the association of keywords, so-called tags, with items of interest in a system. It proved beneficial in systems where a large community of users had to share content in a weakly controlled environment. Nonetheless it was shown that controlled vocabularies often emerge out of communities, which share interest in a topic [5] and we expect a similar behavior from users within our scenario working on joint problems. Further, tagging can dynamically adopt to new technologies or situations, while controlled vocabularies would need large effort in order to adopt them to a steadily changing environment. In our case, tagging specifically eases the *integration of knowledge*. Different sources of information (files, mails, ...) about the same subject can be tagged with same keyword, providing a semantically meaningful link between them. If tags are employed for *knowledge sharing*, a user can define access rights for remote users based on his own taggings and therefore distribute data implicitly. Finally, *knowledge acquisition* is done “on the fly”, since the tags bear important semantics for the user and also within a community of users.

Ontologies. Ontologies are explicit conceptualizations of a shared understanding of domains of interest [8]. They usually model common terms and their relations agreed upon by a community, and can be viewed as controlled vocabularies with further support for reasoning. Ontologies offer important mechanisms for the *integration of knowledge*. By providing a common vocabulary and the relationships between terms, they assist expert users in finding suitable annotations. Moreover, ontologies also allow to infer some higher-level information from basic facts. Further, the *knowledge acquisition* is ameliorated by the use of ontologies, since it provides terminology that expert users are encouraged to share. This vocabulary itself can be considered as a product of collaboration within expert community and based on the taggings provided by particular experts or groups. Therefore, the problem-specific *background knowledge* might evolve, but might also be partly pre-defined, depending on specific requirements of the application domain.

Semantic Desktop. The Semantic Desktop [2] aims at providing a framework for the integration of applications and data based on metadata, so that links are kept between different files, mails and other sources of information. The *integration of knowledge* is one of the main goals of the semantic desktop. It is done in a way transparent to the user, i.e. the metadata is produced from within the usual applications, and not explicitly by additional third-party tools. Semantic desktop enabled applications support *knowledge acquisition* by providing metadata either autonomously or with user interaction. An important aspect is *tracking of provenance* within such applications, i.e. storing information about the source of knowledge, such as the sender of a mail, the author of an document, or reasons for the particular decision.

Peer-To-Peer. A Peer-To-Peer (P2P) system is a decentralized network of computers (e.g. desktop PCs of expert users) without distinction between client and server functionality [7]. In other words, each machine acts both as a server and as

a client, depending on the action that it performs. Data intended for distribution within the P2P network is indexed by sophisticated algorithms that use decentralized data structures and is therefore efficiently available. A P2P infrastructure eases *knowledge sharing* by providing scalable and robust mechanisms for sharing of local knowledge sources with other users. The P2P architecture also offers the *flexible infrastructure* for our framework: no central storage and loose infrastructure of autonomous machines.

RDF. The *Resource Description Framework* RDF [6] is a graph-based model that allows for a flexible representation of metadata. Its schema-less model supports the integration of metadata from different, a priori unknown data sources. With RDF, integrating data sources is reduced to merging sets of triples, and applications only take those statements into account that are significant for them. RDF covers three important requirements as a data representation language. First, it eases the *integration of knowledge* due to its simple, graph based model. Second, it supports the *flexible infrastructure* on the data representation level, and we don't have to adhere to a static schema. Third, *tracking of provenance* can be also supported by RDF. If each item on the personal computer of a user is associated with an URI, provenance information can be easily represented by RDF statements. An important advantage is, that the resulting links become manifest in the data model.

4 Infrastructure

Based on the design decisions mentioned above we envision an architecture (Figure 1) for a networked semantic environment integrating semantic and legacy applications and allowing for easy, transparent exchange of metadata and content. A core element is a local RDF repository on each computer which is used to store metadata. Applications producing semantic metadata like the *semantics aware messenger SAM* directly use the repository. Legacy applications can access the repository via a filesystem interface called *SemFs*. A *semantic exchange architecture SEA* based on P2P technology allows for exchange of content and metadata. In this section we will describe each of these parts of the framework and give a usage example integrating the parts.

4.1 SemFs - A Semantic Filesystem

In order to integrate legacy applications into a semantic desktop we develop *SemFs*, a semantic filesystem, which provides a mapping from modifications of files and directory structures via the traditional filesystem interface to modifications of arbitrary information objects and corresponding annotations in an RDF metadata repository. The annotated information objects are not limited to local files, but can come from any information source, as long as they can be mapped into a file representation at access time.

SemFs translates filesystem paths into queries to functionally nested *views* on the underlying metadata repository. Views are predefined and can be parameterized. For example an 'author'-view could be defined, which would return all

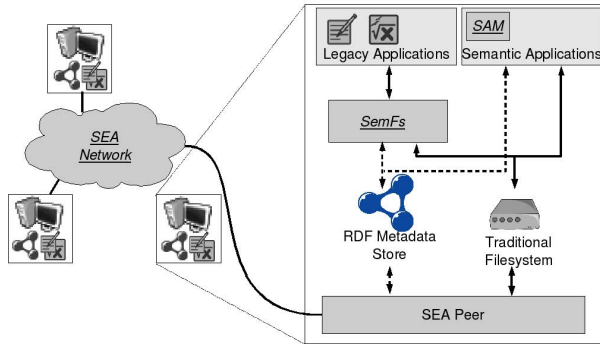


Fig. 1. A Networked Semantic Environment

information objects having the author given as a parameter. The RDF graphs resulting from the filtering operations are mapped to a tabular structure containing names and URLs of the information objects. The names are then used for presenting the information objects as files in the filesystem. The URLs are used to dynamically retrieve the content of the information objects when the corresponding files are accessed. Possible query refinements are presented as subdirectories. The resulting filesystem is purely virtual: All files and directories are generated on-the-fly based on metadata and materialized only when accessed by a client application. When creating a file, placing it inside a directory adds all metadata to the repository, which is necessary to make the file appear within the view corresponding to the directory. Hence SemFs can be used for metadata based browsing as well as for annotation. As an example let assume that a fictional user Becky stores a report about engine number **E55-66-77** in a folder with the same name. Later, another user Chris looks for this report and remembers it was written by Becky. He opens the folder **author: Becky** and finds there all documents written by Becky. He refines his search by changing to the subfolder **E55-66-77** and finally obtains the desired report.

There are two benefits resulting from this flexible architecture: On the one hand we can transparently integrate local and remote information objects through a single, well known interface. On the other hand legacy applications can be provided with access to semantic metadata to a certain extent without having to modify the application. SemFs is described in more detail in [1].

4.2 SEA - A Semantic Exchange Architecture

To allow for autonomously exchange of annotated information objects without a centralized repository we develop the Semantic Exchange Architecture *SEA*. *SEA* provides easy exchange of information objects over a P2P network based on taggings. *SEA* works on the annotation produced by local applications, which is stored in the RDF repository. Based on tags used for annotation the user can define access rights, for example "files tagged with **public** are freely accessible" or "files tagged with **E55-66-77** may be read by the users Adam, Becky and

Chris". Corresponding to the mentioned examples we distinguish three kinds of data: *Private data* has no tags for which access rights are defined. *Publicly shared data* is freely accessible. *Protected data* is accessible by a restricted user group only.

For public data, tagging and location information are published in a structured P2P network using a distributed hash table, which guarantees that information can be found network-wide within a logarithmic number of hops. Protected data is searched for at a finite list of "known" peers, similar to a buddy list for an instant messaging application. This is sufficient, as the user granting access is expected to know the user she grants access to and vice versa. Shared data and annotation can be accessed via a REST API by local applications and remote peers. SEA is described in more detail in [3].

4.3 SAM - A Semantics Aware Messenger

SAM is an instant messaging tool - implemented as a plugin for the Spark² instant messenger - that is based on an ontology-based RDF representation for messaging data. Storing this data in a common repository allows for integration with desktop applications such as SEA or SemFs to improve message management. SAM features collective message tagging, i.e. messages can be annotated by multiple participants of a conversation. If one conversation partner tags a message, the tagging is automatically distributed to other participants of the meeting. The additional context information provided by taggings is used for message retrieval and to enable message collation, e.g. to summarize messages that address some particular problem or that are associated with some project or department. SAM has been implemented for the extensible messaging and presence protocol (XMPP) used by the Jabber network. The protocol has been extended to enable RDF data transfer and data request in form of SPARQL queries. SAM is our first step towards an integrated annotating communication infrastructure, which allows to retrieve knowledge produced during communication from archived conversations. We describe SAM in more detail in [4].

4.4 User Interaction

The infrastructure presented before integrates different data sources and applications to provide an environment for semantic metadata exchange and its collective management. In the following, the user interfaces that build upon this environment are introduced along the lines of the work setting described in Section 2. We consider here the purely fictional use-case scenario of problem analysis and resolution for the airline *ISWebFlights* that reports its trouble to the jet engine manufacturer.

Vision of the Scenario Using ISWeb Tools. The customer service associate Adam gets a phone call of a customer *ISWebFlights* that reports about potentially abnormal sound of one particular jet engine with serial number *E55-66-77*.

² <http://www.jivesoftware.org/spark/>

Adam takes notes about this service request in a text file and uses the conventional *file save* dialog of his preferred text editor to save the file in the folder `ISWebFlights/serviceRequests/E55-66-77`. Adam then uses SAM to quickly write an instant message to the service engineer Becky to ask whether she can inspect the engine. Adam tags this message with `serviceRequests, E55-66-77`. Becky confirms that she can deal with the service request and utilizes the tags made by Adam to retrieve the request using SEA by browsing for the taggings made by Adam. When analyzing the engine, Becky takes pictures of the engine and records the engine's sound. She saves these documents on her harddisk under `E55-66-77` in the subfolders `audio` and `image`. She finds out that some loose filler was the problem of the abnormal engine by investigating the documents in the subfolders of `samples/loose filler`. She adds the existing taggings `loose filler` and `samples` to the instant message and other documents in the folder `E55-66-77` as another sample. Becky replaces the filler, writes a maintenance report that she saves in `E55-66-77/reports`, and writes an instant message to Adam that the engine problem is fixed. Since Adam might have forgotten which service request Becky dealt with, she also tags this message with `E55-66-77`. Upon retrieving the message, Adam forwards it to Chris who is responsible for the customer report and billing. Reading the forwarded message, Chris uses the tag `E55-66-77` that is attached to the original message to retrieve anything he can find for this tag with SEA: He retrieves Adam's service request, images, audio material, and Becky's maintenance report and collates this information to produce the problem resolution report and the bill for *ISWebFlights*.

Browsing for Information. SemFs transparently augments conventional file management interfaces, i.e. users continue to use conventional file management tools such as a file browser or *file open* and *file save* dialogs of legacy applications.

Besides functionality expected by file browsers, SemFs enables faceted file browsing, i.e. the same files can be retrieved through different access paths that resemble different user associations or perspectives.

Besides faceted browsing, the integration of multiple data sources allows to retrieve the same information from within different applications. For instance, a message tagged within SAM can be retrieved using SemFs by browsing for all information objects related to that tag. Furthermore, as messages are automatically associated to users (sender, recipients) one can also use SemFs to browse by user to find the message. On the other hand, from within SAM, file system content associated to a message is accessible.

Organization of Knowledge. Filing on the computer desktop means to deposit information, usually in a file for which a name needs to be created that describes the contained information. The file is then put into a particular folder that is created to summarize related information represented in further files. The purpose of filing is to improve later retrieval of filed information where more structure in form of more sophisticated folder hierarchies can improve the retrieval process. Using the tools presented here, filing has the notion of annotating, i.e. adding metadata about some information object. This organizational paradigm is less

complex than the creation of a folder hierarchy and provides more flexibility for browsing information as indicated before. In many cases, filing is not even required to support later retrieval as new information is already accompanied by useful metadata, e.g. the sender and recipient of a message. Users, however, can still be provided with user interfaces that allow them to add further metadata.

5 Conclusion

The aggregation and exploitation of lifecycle knowledge for complex industrial products is a hard and versatile problem. In this paper, we highlighted the semantic infrastructure technology that will help to achieve the challenging goals of knowledge management: acquisition, integration and sharing of knowledge across sources and platforms, its representation and provenance tracking for sophisticated application scenarios like analysis and resolution of non-trivial technical problems. The proposed networked semantic environment uses locally stored RDF repositories as a backbone for knowledge representation, the semantic exchange architecture SEA for knowledge sharing in a distributed P2P environment, the semantic filesystem SemFs for customizable knowledge representation, and the instant messaging toolkit SAM for expert exchange and background knowledge acquisition.

Our long-term objective is the integrated knowledge management framework for run-through support of knowledge extraction, capturing, representation, and re-use.

References

1. S. Bloehdorn, O. Görlitz, S. Schenk, and M. Völkel. TagFS - Tag Semantics for Hierarchical File Systems. In *International Conference on Knowledge Management*, 2006.
2. S. Decker and M. R. Frank. The Networked Semantic Desktop. In *WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web*, 2004.
3. T. Franz, C. Saathoff, O. Gorlitz, C. Ringelstein, and S. Staab. SEA: A Lightweight and Extensible Semantic Exchange Architecture. In *Proceedings of the 2nd Workshop on Innovations in Web Infrastructure. WWW Conference*, 2006.
4. T. Franz and S. Staab. SAM: Semantics Aware Instant Messaging For the Networked Semantic Desktop. In *Proceedings of the 1st Workshop On The Semantic Desktop. ISWC*, 2005.
5. S. A. Golder and B. A. Huberman. The Structure of Collaborative Tagging Systems. *The Journal of Information Science*, 2006.
6. F. Manola and E. Miller. RDF Primer, February 2004. <http://www.w3.org/TR/rdf-primer/>.
7. S. Staab and H. Stuckenschmidt. *Semantic Web and Peer-to-Peer*. Springer, 2006.
8. S. Staab and R. Studer. *Handbook on Ontologies*. Springer, 2004.

Argument-Based Machine Learning

Ivan Bratko, Martin Možina, and Jure Žabkar

University of Ljubljana, Faculty of Computer and Information Sc.,

Tržaška 25, 1000 Ljubljana, Slovenia

{ivan.bratko, martin.mozina, jure.zabkar}@fri.uni-lj.si

Abstract. In this paper, some recent ideas will be presented about making machine learning (ML) more effective through mechanisms of argumentation. In this sense, *argument-based machine learning* (ABML) is defined as a refinement of the usual definition of ML. In ABML, some learning examples are accompanied by arguments, that are expert's reasons for believing why these examples are as they are. Thus ABML provides a natural way of introducing domain-specific prior knowledge in a way that is different from the traditional, general background knowledge. The task of ABML is to find a theory that explains the "argued" examples by making reference to the given reasons. ABML, so defined, is motivated by the following advantages in comparison with standard learning from examples: (1) arguments impose constraints over the space of possible hypotheses, thus reducing search complexity, and (2) induced theories should make more sense to the expert. Ways of realising ABML by extending some existing ML techniques are discussed, and the aforementioned advantages of ABML are demonstrated experimentally.

Keywords: Machine learning, argumentation, rule learning, CN2, inductive logic programming.

1 Introduction

A fundamental problem of machine learning is dealing with large spaces of possible hypotheses. Usually, this problem is addressed either by language and preference bias. Here, we review some developments with a recent idea for constraining the search space. This idea resembles some notions from the field of argumentation [9]. The idea is based on exploiting arguments about chosen individual examples for learning, rather than on the use of general background knowledge that concerns the whole domain. The use of arguments leads to 'argument-based machine learning', abbreviated as ABML, introduced in [2]. Related ideas were also discussed in [4].

Let us formulate the difference between the usual task of machine learning from examples, and ABML. Usually the problem of learning from examples is stated as:

Given examples

Find a theory that is consistent with the examples

We will now extend this problem statement to that of ABML. In ABML, some of the learning examples are accompanied by arguments, or reasons. Such examples are

called *argued examples*. Argued examples are a means for the domain expert to introduce his (partial) domain knowledge to the learning system prior to learning. The practical advantage of providing prior knowledge in the form of arguments about *individual* examples, in comparison with providing more general domain knowledge, is in that it is easier to explain a single example than giving general theories. As arguments are reasons for a specific example, the expert does not have to ensure that the arguments are true for the whole domain. It suffices that they are true in the context of the given argued example. For instance, consider the problem of deciding whether a credit application is a good one or a risky one. If we ask an expert to explain why was a certain application classified as risky, he or she could explain that this was due to unemployed status of the applicant. This is, naturally, not valid for the whole domain as there are other ways of securing the credit, say offering property as a collateral. But, as the particular applicant did not secure his credit in any other way, the expert states unemployment as a deciding factor, while not mentioning other attributes.

With arguments, the learning problem statement changes to:

Given examples + supporting arguments for some of the examples

Find a theory that explains the examples using given arguments

The main motivation for using arguments in learning lies in two expected advantages:

1. Reasons (arguments) impose constraints over the space of possible hypotheses, thus reducing search complexity
2. An induced theory should make more sense to an expert as it has to be consistent with given arguments.

Regarding advantage 1, by using arguments, the computational complexity associated with search in the hypothesis space can be reduced considerably, and enable faster and more efficient induction of hypotheses. Regarding advantage 2, there are many possible hypotheses that, from the perspective of a machine learning method, explain the given examples sufficiently well. But some of those hypotheses can be incomprehensible to experts. Using arguments should lead to hypotheses that explain given examples in similar terms to those used by the expert, and correspond to the actual justifications.

2 Constructing ABML Algorithms

2.1 Exploiting Argued Examples in ABML

To illustrate the idea of exploiting argued examples in ABML, consider a simple learning problem about credit approval. Each example is a customer's credit application together with the manager's decision about credit approval. Each customer has a name and three attributes: PaysRegularly (with possible values "yes" and "no"), Rich ("yes" or "no") and HairColor ("black", "blond", ...). The class is CreditApproved ("yes" or "no"). Let there be three learning examples as follows:

Name	PaysRegularly	Rich	HairColor	CreditApproved
Mrs. Brown	no	yes	blond	yes
Mr. Grey	no	no	grey	no
Miss White	yes	no	blond	yes

A typical rule learning algorithm might induce the following theory about this domain:

IF HairColor = blond THEN CreditApproved = yes ELSE no

This is short and consistent with the data. Now consider some arguments that may change the situation. The expert's argument about Mr. Grey can be: Mr. Grey did not receive credit because he does not pay regularly. This is a very reasonable argument, however notice that it does not hold for the whole domain. Mrs. Brown, for instance, did receive credit although she does not pay regularly. The expert's argument about Mrs. Brown can be: Mrs. Brown received credit because she is rich, although she does not pay regularly. How should an ABML algorithm react to these arguments? The above rule that uses HairColor as the criterion is not consistent with the arguments. Although it does cover Mrs. Brown in the usual sense (it classifies Mrs. Brown as "yes"), but is not acceptable in the view of the argument for Mrs. Brown: it does not explain her credit is due to her being rich. Instead, an argument based rule learning algorithm would therefore induce the rule:

IF Rich = yes THEN CreditApproved = yes

This rule explains Mrs. Brown example using given arguments. We say that this rule *AB-covers* Mrs. Brown example. The following rule

IF Rich = yes AND PaysRegularly = no THEN CreditApproved = yes

however does *not* AB-cover Mrs. Brown because it is not consistent with the expert's negative argument »although she does not pay regularly«.

2.2 Argument-Based Rule Learning with ABCN2

ABCN2 [7] is an algorithm for learning rules in the argument-based framework for machine learning. ABCN2, which stands for argument-based CN2, is an extension of the well known CN2 rule induction algorithm of Clark and Niblett [3]. CN2 was extended to ABCN2 roughly along the lines described in the previous section. One extension was the refinement of the definition of a rule *covering* an example. In the standard definition in CN2, a rule covers an example if the condition part of the rule is true for this example. In argument based rule learning, this definition is modified to *AB-covering* as illustrated above. The CN2 algorithm [3] consists of a covering algorithm and a search procedure that finds individual rules by performing beam search. The covering algorithm induces a list of rules that cover all the examples in the learning set. The first requirement for ABML is that an induced hypothesis explains the argued examples using given arguments. In rule learning, this means that for each argued example, there has to be at least one rule in the set of induced rules that AB-covers this example. This is achieved by replacing covering in

original CN2 with AB-covering. In addition to simply AB-covering all the argued examples, we would prefer also explaining as many as possible of the non-argued examples by arguments given for the argued examples. Several other modifications of CN2, to obtain ABCN2, are needed. These include the following:

- Each argued example has to be AB-covered by a rule. Therefore ABCN2 repeatedly selects one of the not yet covered argued examples as a seed, and attempts to find a rule that AB-covers this example. Such a rule will have to contain the reasons of at least one of the positive arguments. Therefore the STAR, that is the set of alternative complexes in CN2, is initially set to the positive arguments of the example to be AB-covered.
- Specialize with selectors that are satisfied by argued example only. This ensures the coverage of the seed example by the induced rule.
- Discard all complexes that are consistent with negative arguments. Again, rules must AB-cover argued example, therefore can not be consistent with any of negative arguments.
- Evaluation of rules was improved to account for the size of hypothesis space [5].

2.3 Argument-Based Inductive Logic Programming

It should be noted that, in similar ways as extending CN2 to ABCN2, various other standard ML techniques can be extended to their argument-based variants. For example, extending ILP to AB-ILP may look as follows. The usual ILP problem statement is (in a somewhat simplified form, not mentioning negative examples):

- Given: Examples E + background knowledge BK
- Find a hypothesis H , such that BK and $H \models E$

Hypothesis H explains the examples w.r.t. BK . This can be extended to AB-ILP problem definition as:

- Given: Examples E , annotated by reasons R , and background knowledge BK
- Find hypothesis H , such that BK and $H \models_R E$

This means: hypothesis H explains the examples w.r.t. BK *using reasons* R . The notation $\models_R$ means : derivation of E mentions reasons R . The ILP program HYPER [1] has been extended to its AB variant called AB-HYPER. Although this is a preliminary work, some typical examples of ILP learning, such as learning family relations, show the speed-up of AB-HYPER over HYPER by orders of magnitude, indicating the power of arguments for reduction of hypothesis space. As combinatorial complexity in ILP is so high, it is here that greatest improvements are most natural to expect.

3 Experimental Demonstrations

It should be noted that ABML techniques cannot be evaluated as easily as most of typical ML techniques by simply applying them to benchmark datasets available in the UCI repository for ML. The reason is that in ABML the involvement of an expert

is required to provide arguments. In this section we describe the findings from some experimental applications of ABCN2.

3.1 A Medical Application

In [8] an application of ABCN2 to a medical domain is described. The medical problem there was severe bacterial infections in geriatric population. This problem is quite significant as elderly population, people over 65 years of age, is rapidly growing as well as the costs of treating this population. In that paper, ABCN2 and CN2 were compared, showing that using arguments enabled some improvements of induced models. The performance of ABCN2 was compared also to that of C4.5, Naïve Bayes and Logistic Regression in this domain.

The goal was to build a model from data which would help the physician, at the first examination of the patient, to determine the severity of the infection, and consequently, whether the patient should be admitted to hospital or could be treated as an outpatient. A further goal was to obtain an understandable model, not a black box, to see which parameters play a decisive role. The medical data was gathered at the Clinic for Infectious Diseases in Ljubljana. Patients over 65 with CRP value over 60 mg/l, which indicated a bacterial etiology of the infection, were included only. The patients were observed for 30 days from the first examination or until death caused by the infection. The data includes 40 clinical and laboratorial attributes acquired at the first examination for each of 298 patients (learning examples). The distribution of the class values was the following: 11.4% for 'death = yes', and 88.6% for 'death = no'. The argumentation was done by the physician who was treating the patients and could give the reasons she believed caused death for 32 examples of death cases. CN2, ABCN2 and C4.5 achieved similar classification accuracy while NB and LogR performed significantly worse in this respect. For imbalanced domains, such as this domain, AUC (area under ROC curve) and Brier score (quadratic error of predicted probability) are more relevant measures of success than accuracy. AUC is the probability that for two randomly chosen examples with different classes, the method will correctly decide the class values for these examples (it is not allowed to classify both in the same class, as it knows that they are from different classes). According to AUC and Brier score, ABCN2 significantly outperformed all other learning methods tried.

The question was studied why and how ABCN2 outperformed the other methods in the comparison. A detailed examination indicated that this was due to ABML's ability to reduce the size of hypothesis search space, and as a consequence the evaluations of induced rules became more reliable. The second expected advantage of ABML, that is the interpretability of induced theories, was not confirmed in this domain. The domain expert could not decide whether there was any benefit in this respect from the use of argumented examples.

3.2 A Legal Application

In many areas of law - especially administrative law - many thousands of cases are routinely decided. Often these cases involve the exercise of some discretion, or involve some degree of operationalisation of the concepts used in the legislation, so

that the decision can be based on ascertainable facts rather than the vaguer terms in which the legislation may be couched. In [6], the question was investigated: Can the rules being followed be reconstructed by ML from past cases? In particular, ABCN2 was applied to data concerning a fictional welfare benefit. For this experiment a data set of 2400 cases was used. Each case was described in terms of 64 attributes, twelve attributes being relevant to the decision, and the remaining 52. were irrelevant attributes. The same dataset had been used in previous experiments with ML in a legal domain. ABCN2 performed at least as well as the others in terms of accuracy, which was very high indeed. However, it is not the accuracy of classification that is considered to be most important to meet the requirements of ML application in law: it is the quality, in terms of interpretability, of the rules generated that is crucial. In the initial learning stage, without arguments, the performance of ABCN2 was similar to that of the others. Additionally, the robustness of ABCN2 in the face of erroneous decisions was explored, that is noise in learning data. It was found that ABCN2 is much more robust against noise than original CN2. This is important because many of the administrative law applications to which the technique could be applied exhibit quite large error rates.

3.3 A Business Application

ABCN2 has also been applied to the problem of learning a company's business rules about credit approval to their customers (a large scale demonstration study in the ASPIC project, a European 6th Framework project). In this experiment, the improvements of ABML over “classical” ML were the most dramatic.

There were 5000 examples for learning to decide about a customer's credit history (“good” or “bad”). With classical ML, without arguments (CN2, in our case), 40 rules were induced, whose accuracy was 95%. ABCN2 was applied to the same data, initially without arguments. ABCN2 also features a mechanism for automatically detecting critical cases among the learning examples, that is those that seem to be the most appropriate for an expert to annotate with arguments. Using this mechanism, ABCN2 selected two critical examples for which a domain expert then provided arguments. The inclusion of these enabled ABCN2 to find a rule set with six rules only (in contrast to 40 without arguments), whereby increasing the accuracy to 99.9%.

4 Conclusions

We reviewed some developments with ABML (Argument-Based Machine Learning), a recent idea of using annotated examples for machine learning, where annotations have the form of arguments, or reasons given by a domain expert to selected examples. Arguments provide a way of introducing prior domain knowledge into the learning process. However, in this case such domain knowledge is concerned with individual examples rather than with the whole learning domain as is the case with more traditional use of background knowledge in machine learning. The advantage of annotating individual examples is that it is typically much easier for a domain expert to state arguments for individual concrete cases than to state general rules. We showed some ideas for exploiting arguments thereby extending ML techniques to

their argument-based variants. The expected benefits of ABML w.r.t. traditional ML are: first, reduction of search space among possible hypotheses, and second, the resulting theories have better chances of making sense to a domain expert. We presented experimental applications that support these expectations.

Acknowledgement. The work reported in this paper was partly supported by the European Commission (6th Framework Project ASPIC), and the Slovene research agency ARRS. We would also like to thank our partners in the ASPIC project.

References

1. Bratko, I.: Prolog Programming for Artificial Intelligence, third edition. Pearson Education / Addison-Wesley, 2001.
2. Bratko, I., Možina, M.: Argumentation and machine learning. Chapter 5 of *Theoretical framework for argumentation*, ASPIC project, deliverable D2.1, 2004.
3. Clark, P., Niblett, T.: The CN2 induction algorithm' *MachineLearning Journal*, **4** (1989) 261–283.
4. Gomez, S.A., Chesnevar, C.I.: Integrating defeasible argumentation and machine learning techniques. Technical report, Universidad Nacional del Sur, Bahia Blanca 2004.
5. Možina, M., Demšar, J., Žabkar, J., Bratko, I.: Why is rule learning optimistic and how to correct it? *Proc. ECML'06*, Berlin 2006.
6. Možina, M., Žabkar, J., Bench-Capon, T., Bratko, I.: Argument Based Machine Learning Applied to Law. *Artificial Intelligence and Law*, 2006 (in press).
7. Možina, M., Žabkar, J., Bratko, I.: Argument based rule learning, *Proc. ECAI'06*, Riva del Garda, Italy, Springer 2006.
8. Možina, M., Žabkar, J., Videčnik, J., Bratko, I.: Argument based machine learning in a medical domain, *Proc. COMMA'06*, Manchester 2006.
9. Prakken, H., Vreeswijk, G.: *Handbook of Philosophical Logic, second edition*, volume 4, chapter Logics for Defeasible Argumentation, 218–319, Kluwer Academic Publishers, Dordrecht etc, 2002.

Play It Again: A Case-Based Approach to Expressivity-Preserving Tempo Transformations in Music

Ramon López de Mántaras

IIIA - Artificial Intelligence Research Institute
CSIC - Spanish Council for Scientific Research
mantaras@iiaa, csic.es

Abstract. It has been long established that when humans perform music, the result is never a literal mechanical rendering of the score. That is, humans deviate from the score. As far as these performance deviations are intentional, they are commonly thought of as conveying musical expressivity which is a fundamental aspect of music. Two main functions of musical expressivity are generally recognized. Firstly, expressivity is used to clarify the musical structure of the piece (metrical structure, phrasing, harmonic structure). secondly, expressivity is also used as a way of communicating, or accentuating, affective content.

An important issue when performing music is the effect of tempo on expressivity. It has been argued that temporal aspects of performance scale uniformly when tempo changes. That is, the durations of all performed notes maintain their relative proportions. This hypothesis is called relational invariance (of timing under tempo changes). However, counter-evidence for this hypothesis has been provided, and a recent study shows that listeners are able to determine above chance-level whether audio recordings of jazz and classical performances are uniformly time stretched or original recordings, based solely on expressive aspects of the performances. In my talk I address this issue by focusing on our research on tempo transformations of audio recordings of saxophone jazz performances. More concretely, we have investigated the problem of how a performance played at a particular tempo can be automatically rendered at another tempo while preserving its expressivity. To do so we have developed a case-based reasoning system called TempoExpress. Our approach also experimentally refutes the relational invariance hypothesis by comparing the automatic transformations generated by TempoExpress against uniform time stretching.

Decision Fusion of Shape and Motion Information Based on Bayesian Framework for Moving Object Classification in Image Sequences

Heungkyu Lee¹, JungHo Kim², and June Kim³

¹ Dept. of Computer Science, Seokyeong University, Seoul, Korea

² Republic of Korea Navy 1st Fleet

³ Dept. of Information and Communication Engineering
Seokyeong University, Seoul, Korea

hklee@skuniv.ac.kr, jhkim@ispl.korea.ac.kr,

june@skuniv.ac.kr

Abstract. This paper proposes decision fusion method of shape and motion information based on Bayesian framework for object classification in image sequences. This method is designed for intelligent information and surveillance guard robots to detect and track a suspicious person and vehicle within a security region. For reliable and stable classification of targets, multiple invariant feature vectors to more certainly discriminate between targets are required. To do this, shape and motion information are extracted using Fourier descriptor, gradients, and motion feature variation on spatial and temporal images, and then local decisions are performed respectively. Finally, global decision is done using decision fusion method based on Bayesian framework. The experimental results on the different test sequences showed that the proposed method obtained good classification result than any other ones using neural net and other fusion methods.

1 Introduction

Recently, development of intelligent surveillance guard robot is actively in progress to operate guard tasks to detect and classify moving object using CCD camera in military, national major infra-facilities and so on. That is why much manpower is expended on sentry duty to guard the premises against unauthorized personnel for 24 hours for surveillance tasks everyday. To lessen the time and effort expended by human guard, an intelligent surveillance guard robot is desirable.

For doing this, we design and develop the detection, tracking, and classification method [1]. In this paper, we limit the scope to the target detection and classification method. We confine that classification classes are three such as human, car, and animal having front-view, side-view and inclined-view. These classes have rigid and non-rigid characteristics of shape information. Thus, for reliable and stable classification of targets, multiple invariant feature vectors to more certainly discriminate between targets are required. To do this, shape and motion information are extracted using Fourier descriptor, gradients, and motion feature variation [1][2][6] on spatial and temporal images, and then local decision is performed. Finally, global decision is

done using decision fusion based on Bayesian framework. This method provides effective method for the combined decision of respective local feature analysis obtained from shape and motion information. In addition, surveillance range is from 5m to 1km. For doing this, zoom lenses are used for zoom-in and zoom-out, and minimum recognizable blob size is determined by 10×10 pixels for moving object classification.

First, we use two-stage adaptive background subtraction algorithms for extraction of moving objects from a video sequence. These algorithms have the advantage of increasing the accuracy of the foreground segmentation [2][3][4][5]. Once the foreground is extracted, proposed system consists of three feature extraction procedures: First, Fourier descriptor and gradients representing the shape that is invariant to several change such as translation, rotation, scale, and starting point, is computed, and then classification task for local decision is performed using neural network. Second, classification task for local decision using temporal information is performed using motion information to be obtained from rigidity condition analysis of moving objects. For doing this, skeletonization of the motion region is done, and then motion analysis is done to compute motion feature variation using selected feature points [7][8].

Finally, we can classify moving objects through decision fusion method based on Bayesian framework using locally obtained results from shape and motion analysis [9][10]. Figure 1 shows an overview of the overall system.

This paper is organized as follows. In Section 2, we describe multiple invariant feature vectors from spatio-temporal image. In Section 3, we describe the decision fusion method based on Bayesian framework. In Section 4, we evaluate experimental results compared to other methods with representative simulations. Finally, concluding remarks are presented in Section 5.

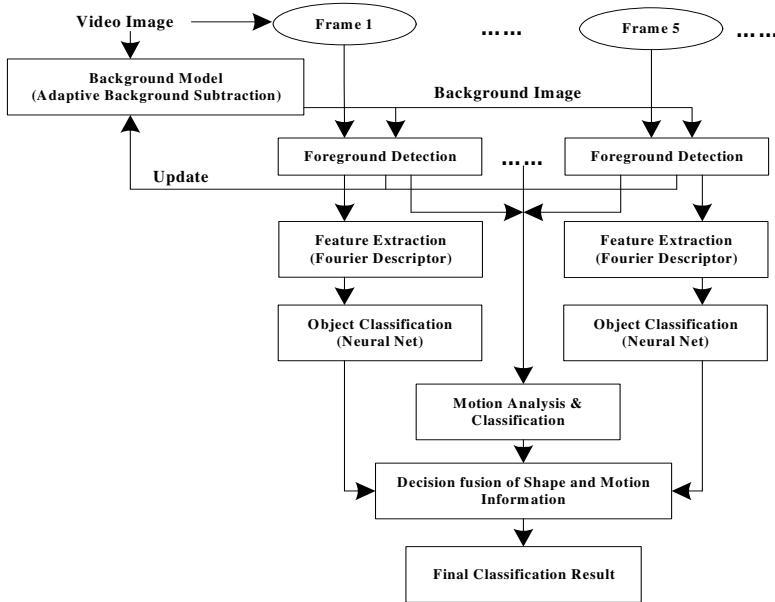


Fig. 1. Overall system architecture

2 Multiple Invariant Feature Vectors from Spatio-temporal Space

2.1 Fourier Descriptors from Spatial Image

The use of Fourier descriptors is common in pattern recognition and image analysis. The benefits of the Fourier descriptors are invariance to the starting point of the boundary and rotation. Fourier descriptor representing shape information can be computed based on the boundary of the object as follows. First, distances from object center points to the boundary points are computed as

$$r(k) = ((x(k) - x_c)^2 + (y(k) - y_c)^2)^{1/2}, \quad k = 0, 1, \dots, K-1 \quad (1)$$

that is translation invariant. Here, K is the length of boundary and (x_c, y_c) is the center of the object. Using this distance function, the boundary can be represented independently on the location of the object in the image. Fourier descriptors of a distance function of boundary can be obtained using the discrete Fourier transform (DFT). Fourier descriptors characterize the object shape in frequency domain.

$$a(u) = \frac{1}{N} \sum_{k=0}^{K-1} r(k) e^{-j2\pi uk/K}, \quad u = 0, 1, \dots, K-1 \quad (2)$$

Here, the coefficients $a(u)$ are called by the Fourier descriptors of the boundary. The equation (3) can be preprocessed for obtaining rotation invariance as follows:

$$fd = \left[\frac{|a(u)|}{|a(0)|} \right], \quad u = 1, 2, \dots, 2/K \quad (3)$$

The general shape of the object is represented by the lower frequency descriptors and high frequency descriptors representing detailed shape information of objects. In this paper we use only the first 30 coefficients for the object classification task. This quantity was derived empirically after consideration of classification rate and speed.

2.2 Gradients from Temporal Image

We compute first derivative to detect the discontinuity of image intensity from detected object boundary. The gradient of two-dimensional function, $N(x, y)$ can be defined as

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial N(x, y)}{\partial x} \\ \frac{\partial N(x, y)}{\partial y} \end{bmatrix} \quad (4)$$

Gradient values is zero in a region having uniform intensity, meanwhile, variable intensity is proportional to the image brightness. Basic characteristic of gradient vectors indicates the maximum rate of variation in direction from coordinate, (x, y) to N as follows:

$$\tan \theta = \frac{\frac{\partial N(x, y)}{\partial y}}{\frac{\partial N(x, y)}{\partial x}} = \frac{\Delta y}{\Delta x} \quad (5)$$

$$\theta = \arctan\left(\frac{\Delta y}{\Delta x}\right)$$

From equation (5), each theta for each pixels of boundary image can be used for computation of pixel counts according to the vertical and horizontal components of gradient as follows:

$$\theta_{ver} = \begin{cases} \text{add count,} & |\theta| > Th_{ver} \\ \text{keep count} & \text{else} \end{cases} \quad (6)$$

$$\theta_{hor} = \begin{cases} \text{add count,} & \theta < Th_{hor} \\ \text{keep count} & \text{else} \end{cases} \quad (7)$$

From equation (6) and (7), we can use ratio of vertical and horizontal angle as a feature vector. This is location, scale, rotation invariant because it is obtained from object boundary. For example, this feature can be used for the local decision in two class classification problem as follows:

$$Z_k = \begin{cases} \text{human} & \text{if } \theta_{ver} / \theta_{hor} < Th \\ \text{vehicle} & \text{else} \end{cases} \quad (8)$$

2.3 Motion Feature Variation from Temporal Image

For analysis of motion feature variation, first we compute the skeleton of object. There are many standard techniques for skeletonization such as thinning, distance transformation, and “star” skeleton [8]. In this paper, we used “star” skeleton method to detect external points on the boundary of the motion region. The advantage of this method is low computational cost, and provides a mechanism for controlling scale sensitivity. In addition, it relies on no a priori object model. This skeleton procedure is shown in Figure 2 and can be described as follows:

1. The center of gravity of the moving region boundary is determined,
2. The distances from the center of the region to each boundary point are calculated.
The signal is then smoothed using low-pass filter for noise reduction.
3. Local maximal points are obtained from the filtered signal. These points are considered as extremal points of moving region boundary.
4. Finally, the skeleton of moving region is obtained by connecting extremal points to the center of gravity of the moving region boundary.

We selected left-most lower extremal points such as a leg (human and animal) or bumper (car) as feature points of skeleton. Variation of these points is higher than other extremal points. If X is the set of extremal points, left-most lower extremal point (l_x, l_y) is obtained according to the below condition (9).

$$X = \{(x_i, y_i)\}, \quad i = 1, \dots, n$$

$$(l_x, l_y) = (x_i, y_i) = \min_{y_i < y_c} x_i \quad (9)$$

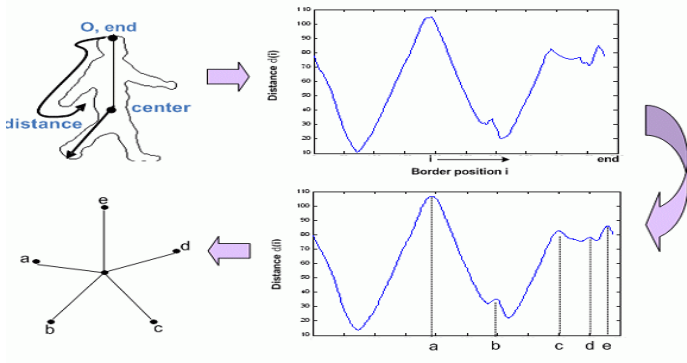


Fig. 2. The region skeleton is obtained from the distances from the center of the region to each boundary point

These feature point of objects (human, car, and animal) are selected for motion feature variation as shown in Figure 3. Using these points, we compute the angle θ (l_x, l_y) that makes with the vertical as follows:

$$\theta = \tan^{-1} \frac{l_x - x_c}{l_y - y_c} \quad (10)$$

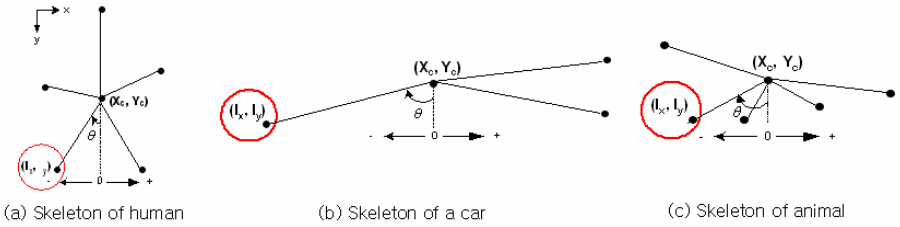


Fig. 3. Feature points of moving objects

3 Decision Fusion Method in Bayesian Framework

In this paper, decision problem is considered as binary hypothesis testing to classify the given features into human, vehicle, and animal. From multivariate feature vectors, respective local decisions are made. And then, we apply the global fusion rule to combine local decisions $u_i, i=1,2,3$ based on some optimization criterion for global decision u_0 . We derive the optimum fusion rules that minimize the average cost in a Bayesian framework [9][10]. This rule is given by the following likelihood ratio test:

$$\begin{aligned}
& u_0 = 1 \\
& \frac{P(u_1, u_2, u_3 | H_1)}{P(u_1, u_2, u_3 | H_0)} > \frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})} \cong \eta \\
& u_0 = 0
\end{aligned} \tag{11}$$

where C_{ij} is the cost of global decision. The left-hand side can be rewritten as given in equation (12) because the local decisions have characteristic of independence.

$$\begin{aligned}
\frac{P(u_1, u_2, u_3 | H_1)}{P(u_1, u_2, u_3 | H_0)} &= \prod_{i=1}^3 \frac{P(u_i | H_1)}{P(u_i | H_0)} \\
&= \prod_{S_1} \frac{P(u_i = 1 | H_1)}{P(u_i = 1 | H_0)} \prod_{S_0} \frac{P(u_i = 0 | H_1)}{P(u_i = 0 | H_0)}
\end{aligned} \tag{12}$$

where S_j is the set of all those local decisions that are equal to j , $j=0,1$. In addition, equation (12) can be rewritten in terms of the probabilities of false alarm and miss in detector i . That is why each input to the fusion center is a binary random variable characterized by the associated probabilities of false alarm and miss.

$$\prod_{S_1} \frac{P(u_i = 1 | H_1)}{P(u_i = 1 | H_0)} \prod_{S_0} \frac{P(u_i = 0 | H_1)}{P(u_i = 0 | H_0)} = \prod_{S_1} \frac{1 - P_{Mi}}{P_{Fi}} \prod_{S_0} \frac{P_{Mi}}{1 - P_{Fi}} \tag{13}$$

where $P_{Fi} = P(u_i = 1 | H_0)$ and $P_{Mi} = P(u_i = 0 | H_1)$. We substitute equation (13) in equation (11) and take the logarithm of both sides as follows:

$$\begin{aligned}
& u_0 = 1 \\
& \sum_{S_1} \log \frac{1 - P_{Mi}}{P_{Fi}} + \sum_{S_0} \log \frac{P_{Mi}}{1 - P_{Fi}} > \log \left(\frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})} \right) \cong \log \eta \\
& u_0 = 0
\end{aligned} \tag{14}$$

The equation (14) can be expressed as shown in equations (15) and (16).

$$\begin{aligned}
& u_0 = 1 \\
& \sum_{i=1}^3 \left[u_i \log \frac{1 - P_{Mi}}{P_{Fi}} + (1 - u_i) \log \frac{P_{Mi}}{1 - P_{Fi}} \right] > \log \eta \\
& < \log \eta \\
& u_0 = 0
\end{aligned} \tag{15}$$

$$\begin{aligned}
& u_0 = 1 \\
& \sum_{i=1}^3 \left[\log \frac{(1 - P_{Mi})(1 - P_{Fi})}{P_{Mi} P_{Fi}} \right] u_i > \log \left[\eta \prod_{i=1}^3 \left(\frac{1 - P_{Fi}}{P_{Mi}} \right) \right] \\
& < \log \left[\eta \prod_{i=1}^3 \left(\frac{1 - P_{Fi}}{P_{Mi}} \right) \right] \\
& u_0 = 0
\end{aligned} \tag{16}$$

Thus, the optimum fusion rule is applied by forming a weighted sum of the incoming local decisions, and then comparing it with a threshold. At this time, the threshold depends on the prior probabilities and the costs.

4 Experimental Results

The algorithm for moving object classification through fusion of shape and motion information presented in this paper has implemented, and then run under Window XP

operating system. For experimental evaluation, training images are obtained in real image sequences. Each class includes three kinds of image view having front, side, and inclined image. Each kinds of image are composed of total 1200 files respectively for training, which is extracted from respective image sequences. Test images were also obtained from image sequences (image size is 480×320) about three targets (human, car, and animal): human (total 400 frames), car (total 400 frames), and animal (total 400 frames) which is a gray level image.

We assume that we found the moving blobs from image sequences, and performed data alignment in image coordinates. For experimental evaluation, adaptive background detection [3] and time difference method between adaptive background model and captured current image are applied to obtain moving image blobs. From this result, binary thresholding and morphology filter, closing are applied. Then, each obtained moving blob is labeled. For representing the obtained targets, we present a set of minimum bounding rectangles (MBR) employing “object range” and “validation region”, as a means to represent the position, size and region of a target as feature vectors for describing the accuracy bound and range. It is important for any data representation to contain a measure of the quality of data. If a sensor is operating reasonably well, it is expected that the acquired data will be within the explicit accuracy bounds. The issue to be considered in out-door field is illumination condition. This condition make the moving blob detection difficult because illumination variation is fast, and this result is represented as moving blobs. Thus, we applied the median filter in computing adaptive background model. In addition, specific part that variation rate of intensity change is high, is not considered as moving blobs.

Once moving blobs are detected, object boundary is computed using edge following algorithm as shown in Figure 4, and then feature vectors are computed. First, Fourier descriptors are computed, and then ratio of horizontal and vertical angle is computed on spatial image frame as described in Section 2. Second, motion feature variation is computed from temporal image sequences using object skeleton as shown in Figure 5. Each feature vector is inputted for local decision using neural network respectively. And then, local decisions are combined using global fusion method based on Bayesian framework. Multiple local decisions can be the important factor to do reliable decision-making. In addition, each local decision provides the complementary information for reliable decision-making. Thus, global fusion method derives the reliable decision making method.

To show the robustness of proposed method, we evaluated the proposed method which is compared to other methods such as neural net, and some fusion methods [12][13][14]. Majority voting and weighted average score method are fusion methods that are compared to the proposed fusion method. To show the robustness of the



Fig. 4. Object boundary extraction procedure after detecting moving boundary

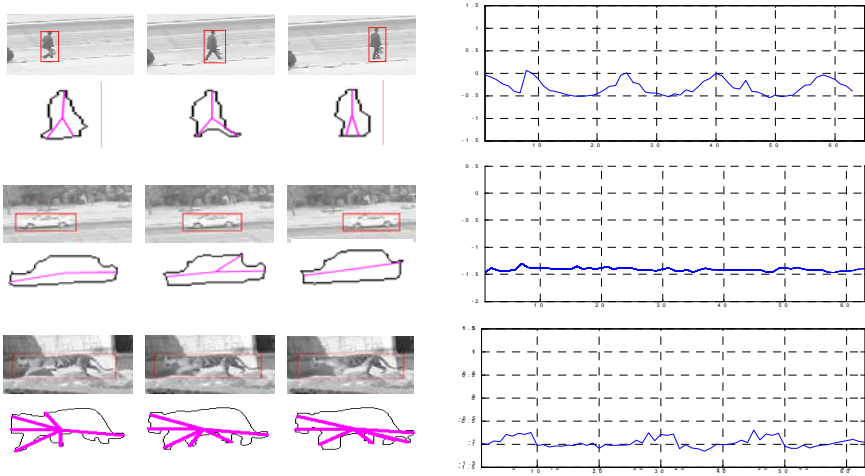


Fig. 5. Variation of the angle θ in side view image sequences (human, car, and animal)

proposed method, we compared FAR (False Acceptance Rate), FRR (False Rejection Rate), and classification rate of each method. Table 1 shows the experimental results of three methods respectively. Local decision method using neural network showed the lowest classification rate. Each local decision did not show satisfactory results. Thus, some fusion methods are secondly compared using respective local decisions. From the results, majority voting and weight average score methods showed low performance compared to the proposed method considering spatio-temporal information. Thus, we can know that the simple fusion method of final decision does not bring the good performance. In addition, we can know that the fusion method of redundant and complementary information is good choice in feature level and decision level fusion.

Table 1. Experimental evaluations compared to previous methods

Method	FRR(%)	FAR(%)	Recognition Rate(%)
Neural Net	4.0	3.0	96
Majority Voting	2.7	2.5	97.3
Weight Average Score	2.5	2.0	97.5
Proposed Method	1.5	1.3	98.5

5 Final Remarks and Conclusions

The proposed scheme is designed for intelligent information and surveillance robots using one fixed camera and one active camera as shown in Figure 6. The fixed camera monitors the fixed range, and zooms in/out per arbitrary time or by using manual method to detect only moving targets. Meanwhile, the active camera is used for target tracking. So, it includes functions of zoom in/out, and rotation. For target recognition or classification, the active camera provides the detected moving blob images

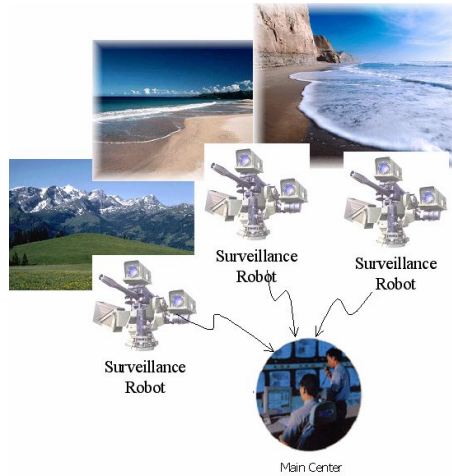


Fig. 6. Applications of surveillance guard robots

to the target classifier. Finally, the target classifier has only to classify the detected blobs into specific classes. However, intruders usually dress in camouflage fatigues and move under false colors. In addition, they have characteristics of non-rigidity. Thus, it is actually difficult to discern his identity. Meanwhile, vehicles such as truck, tank and so on, are somewhat easy target to detect and discern its class as well as are having characteristic of rigidity. From the above situation analysis, we lessened the number of target class to be classified. In place of that, we applied the semi-automatic method to operate the surveillance center, and classify the broad style's targets.

From the experimental evaluation, the proposed method showed the robustness of classification performance than any other methods. However, there are still some methods to be supplemented, which are false alarms such as numerous moving of animals, natural situations occurred from the effects of wind, snow, rain and so on. Thus, preprocessing algorithms to remove false alarms or reliable classify classification algorithms should be studied. On the other hand, some heterogeneous fusion method can be utilized. In our case [11], we have been studying on acoustic detection method using multiple microphone arrays because this method can provide the complementary and redundant information. By using above some approaches, we will enhance the performance of surveillance guard robots. In the near future, we expect that more reliable surveillance guard robots will be appeared.

References

- [1] R. T. Collins, A. J. Lipton and T. Kanade, "Introduction to the special section on video surveillance," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 22, pp. 745~746, Aug 2000.
- [2] A. J. Lipton, H. Fujiyosi, and R. S. Patil, "Moving target classification and tracking from real-time video," *Proc of IEEE Workshop. on Applications of Computer Vision*, pp.8~14, 1998.

- [3] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," Proc of IEEE Conf on Computer Vision and Pattern Recognition, Vol. 2, pp. 246~252, 1999.
- [4] J. Barron, D. Fleet and S. Beauchemin, "Performance of optical flow techniques," Proc of Int. J. Comput. Vis., Vol. 12, No. 1, pp.42~77, .
- [5] I. Haritaoglu, D. Harwood and L. S. Davis, " A real-time system for detecting and tracking people," Proc of Int. Conf. on Face and Gesture Recognition, pp.222~ 227, April 1998.
- [6] Y. Kuno, T. Watanabe, Y. Shimosakoda and S. Nakagawa, "Automated detection of human for visual surveillance system." Proc. of Int. Conf. on Pattern Recognition, pp.865~869, 1996.
- [7] R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis, and applications," IEEE Trans. Pattern Anal. Machine Intell., Vol.22 pp.781~796, Aug. 2000.
- [8] H. Fujiyosi and J. A. Tomasik, "Real-time human motion analysis by image skeletonization", IEICE Trans, on Info & Systems, Vol. E87-D, No. 1, Jan 2004.
- [9] M. M. Kokar, and J. A. Tomasik, "Data vs. decision fusion in the category theory framework", Proc. 2nd Int. Conf. on Information Fusion, 2001.
- [10] Li. X. R., Zhu, Y., Wang, J., Han, C., " Optimal linear estimation fusion-Part I: Unified fusion rules", IEEE Trans. Information Theory. Vol. 49, No. 9, Sep 2003.
- [11] Heungkyu Lee, Jounghoon Beh, June Kim, and Hanseok Ko, "Model based Abnormal Acoustic Source Detection Using a Microphone Array" Advances in Artificial Intelligence, pp.966-969, LNAI 3809, Springer, 2005.
- [12] Y. Bar-Shalom and X. R. Li, Multitarget-multisensor tracking: principles and techniques, YBS Press, 1995.
- [13] Samuel Blackman, Robert Popoli, "Design and Analysis of Modern Tracking Systems", Artech House, 1999.
- [14] R. R. Brooks and S. S. Iyengar, *Multi-Sensor Fusion: Fundamentals and Applications with software*, Prentice Hall, 1998.

A Two-Stage Visual Turkish Sign Language Recognition System Based on Global and Local Features

Hakan Haberdar and Songül Albayrak

Yıldız Technical University,
Computer Science and Engineering Department,
34349 Beşiktaş, İstanbul, Turkey
hakan@haberdar.org, songul@ce.yildiz.edu.tr

Abstract. In order to provide communication between the deaf-dumb people and the hearing people, a two-stage system translating Turkish Sign Language into Turkish is developed by using vision based approach. Hidden Markov models are utilized to determine the global feature group in the dynamic gesture recognition stage, and k nearest neighbor algorithm is used to compare the local features in the static gesture recognition stage. The system can perform person dependent recognition of 172 isolated signs.

1 Introduction

The aim of sign language recognition is to design systems to translate sign language gestures into words. Sign language is characterized by manual and non-manual features [1]. The non-manual features are facial expressions and body direction. The manual features are divided into two categories: global features (location and movement of hands) and local features (hand shape and hand orientation). Using the global and local features together is required to design robust recognition systems.

Sign language recognition systems have begun to appear in the literature since 1990's. To date, device based systems [2] and vision based systems [1,3] are the two main approaches for the recognition task. The device based systems use complex wired data gloves and position trackers to obtain the features. Signer must wear these uncomfortable input devices, whereas the use of a vision based system offers the advantage of common and widely available input device [4]. The vision based systems use one or more video cameras to extract the features of the sign, and the signer may wear simple colored gloves, or skin tone characteristics are used to find the hands and face in the video frames. In the vision based approach, researchers propose static and dynamic gesture recognition methods. The static gesture recognition deals only with static hand posture images while the dynamic gesture recognition utilizes motion information of hands. The hand posture (dynamic gesture recognition) and motion information (static gesture recognition) must be used together to design large lexicon sign language recognition systems.

Charaphayan and Marble [5] proposed an image processing method to recognize correctly 27 of 31 American Sign Language gestures. Starner's famous [3] American Sign Language system could recognize 40 gestures by using distinctly colored gloves. Assan and Grobel [1] presented signer dependent recognition system that is capable of recognizing 262 Netherlands Sign Language gestures with 91.3% accuracy by using colored gloves. In the sign language recognition literature, there are recognition systems for American Sign Language [3], Chinese Sign Language [2], Polish Sign Language [6], and Arabic Sign Language [7] etc. Unfortunately there is not a comprehensive work on Turkish Sign Language. Haberdar and Albayrak [8] proposed the unique Turkish Sign Language recognition system with a vocabulary of 50 gestures, and the recognition accuracy of isolated gestures is 95.7%.

This paper describes a vision based two-stage system developed for the recognition of Turkish Sign Language signs. It analyzes motion and posture information of hands to determine the meaning of the performed sign. The signer is not supposed to wear any kinds of colored gloves because a skin detection method proposed in [9] is used to segment the face and hands from the background. In our previous work, it has been seen that using only four frames, called key frames, is sufficient to recognize the sign word instead of using all the frames [8]. Hidden Markov model, which has demonstrated its ability of dynamic gesture recognition, is utilized to work with the global features in the first stage. The local features are used with k nearest neighbor algorithm to make final decision in the second stage.

The rest of this paper is organized as follows. Information about Turkish Sign Language is given in Section 2. We present the proposed framework and feature extraction in Section 3 and 4 respectively. We show the experimental results in Section 5, and then Section 6 concludes the paper.

2 Turkish Sign Language and the Global Feature Groups

Information about the history of Turkish Sign Language is limited since it is a visual language and thus was hard to put in print [10]. The only book about gestures in Turkish Sign Language is a manual containing 2000 signs published by the Ministry of Education [11]. Non-manual features like facial expressions may modify meaning of the signs, but we discard non-manual effects. Since there is not a national sign database for Turkish Sign Language, we have to learn and perform the signs.

Location and movement of the hands during the sign define the global features, and the hand shape and hand orientation define the local features. All the signs at [10] are carefully examined, and 172 of Turkish Sign Language signs are selected to test the recognition performance of the system.

We arrange 172 signs into 45 distinct groups, called global feature groups (Fig. 1), according to the location and trajectories of the hands (the global features) during performing the signs [12]. The global feature groups give huge

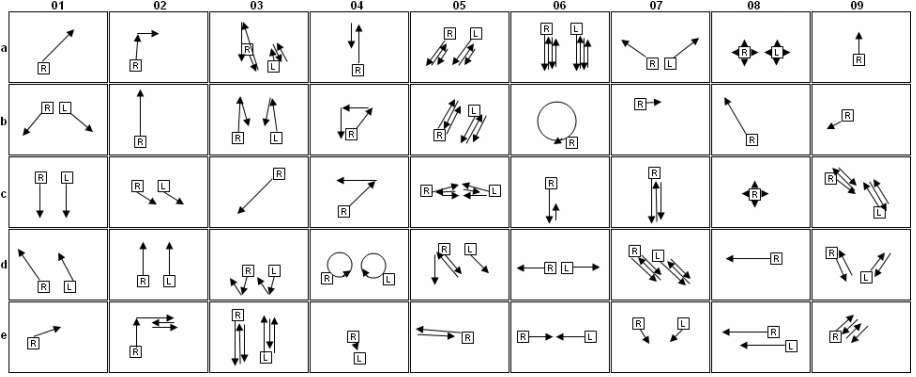


Fig. 1. The global feature groups: a1,..., a9; b1,..., b9; c1,..., c9; d1,..., d9; e1, ..., e9

information about the similarities of the signs. In Fig. 1 R refers to the right hand and L refers to the left hand.

There are different numbers of members in 45 global feature groups. For instance, there is only one sign in the group a2 while there are 26 signs in the group a8. Since all signs in a global feature group have the same global features (for example, all the signs in group a1 are performed by only right hand and the trajectory of the hand is the same for all the group members), we need to use the local features (shape information of hands) to distinguish members in a group.

3 System Framework

The system has 2 stages: global feature group extraction stage (regarding dynamic gesture recognition) and local feature processing stage (regarding static gesture recognition). No input device (such as wired glove or trackers) or colored glove are required for the interface between signer and computer. A charge-coupled device (CCD) camera is utilized to track hands by using the skin detection method in [9].

In the first stage, after face and hands are segmented in consecutive video frames, we select only four frames, called key frames, to represent characteristics of the whole sign. Location and movement information of hands in the four key frames is utilized with 45 Hidden Markov models (each model refers to one global feature group) to determine which global feature group the performed sign belongs to.

In the second stage, if there is more than one sign in the selected global feature group, the local features (hand shape) are used to obtain the corresponding member to the performed sign. k nearest neighbor algorithm is operated to compare differences between the performed sign and previously selected candidates in the global feature group.

3.1 Determining the Global Feature Group

The decision making of the first stage employs Hidden Markov models to determine the global feature group of the performed sign. Although the order of signs in Turkish Sign Language is not a first order Markov process, we assume it to simplify the model. Left-to-right [13] Hidden Markov model, which contains only 4 states including the start and final states that do not emit observations, is used in the system (Fig. 2).

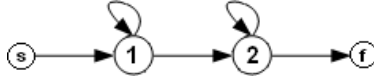


Fig. 2. The Hidden Markov model topology used in the system

A Hidden Markov model consists of 3 problems called evaluation, decoding, and estimation. The evaluation problem deals with the probability of observing an observation sequence when the Hidden Markov model parameters and the state sequence are given. The decoding problem concerns to find the state sequence having the highest probability of generating a given observation sequence. The estimation problem intends to learn the model parameters that maximizes the probability of generating a training set of observation sequences. Detailed information about solutions to the problems are given in [13,14].

The feature vectors that constitute the observation sequences in motion recognition applications are usually continuous. Typically, a vector quantization method can be used to convert continuous value to discrete index. However, the actual probability density functions for the observations may be used instead of using vector quantization. Let $b_j(o_t)$ be the probability density function that gives us the probability of observing a continuous-valued observation o_t in state j at time t . We assume that the observation probability densities are Gaussian with different means and variances:

$$b_j(o_t) = \frac{1}{\sqrt{(2\pi)^{|\Sigma_j|}}} e^{-\frac{1}{2}(o_t - \mu_j)' \Sigma_j^{-1} (o_t - \mu_j)} \quad (1)$$

where μ_j is mean vector and Σ_j is covariance matrix of state j . Initial values for these parameters are calculated by making some approximate assignment of feature vectors to states [14].

In our previous work [8], we see that using absolute positions of hands relative to the coordinate system as shown in Fig. 3 (a) sometimes causes errors, so we define a new coordinate system with a new origin set to the center of the rectangle which covers the face region (Fig. 3 (b)).

Location of the right and left hand in the first key frame and relative position changes in the following 3 key frames constitute our global features (observations) as shown below:

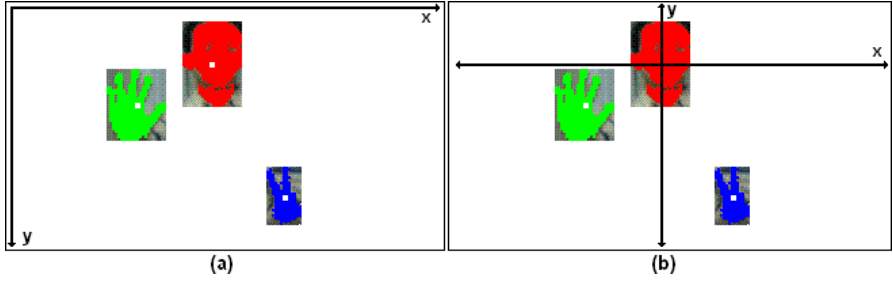


Fig. 3. (a) Frame based coordinate system (b) Face centered coordinate system

$$f = \begin{bmatrix} x_R^1 & y_R^1 & x_L^1 & y_L^1 \\ \Delta x_R^2 & \Delta y_R^2 & \Delta x_L^2 & \Delta y_L^2 \\ \Delta x_R^3 & \Delta y_R^3 & \Delta x_L^3 & \Delta y_L^3 \\ \Delta x_R^4 & \Delta y_R^4 & \Delta x_L^4 & \Delta y_L^4 \end{bmatrix} \quad (2)$$

Observations obtained from motion of the hands during the sign, the global features are given as input of Hidden Markov models, and the global feature group of the performed sign is determined according to output probabilities of Hidden Markov models. The forward algorithm [13] is used to evaluate the probability that a performed sign is generated by the model corresponding to global feature group i . The sign is assigned to the global feature group having the highest probability.

3.2 Local Feature Processing

When there is more than one sign (candidate) in the selected global feature group, as a second stage, the local features are needed to make the final decision. Fig. 4 shows the posture of hands in the key frames.

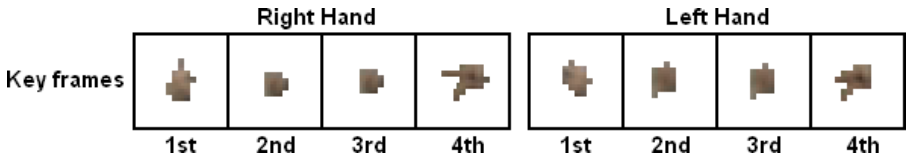


Fig. 4. Hand postures in the key frames

Our experiments show that hand postures in the second and third key frames do not contain distinguished features. Therefore, we used only postures in the first and the last key frames to compare the hand shapes. We use gray level images normalized with respect to size (28 x 28 pixels). The differences among the performed posture and the candidates in the selected global feature group are measured by Euclidean distance [13] of gray pixel values.

If we assume that there are M signs in the selected global feature group and T samples for each sign in the group ($M \times T$ samples) as training data, we can find differences among the test sign and the training samples by simply using Euclidean distance of gray pixel values.

In the equations below, $P_{R1}^o(i, j)$, $P_{R4}^o(i, j)$, $P_{L1}^o(i, j)$ and $P_{L4}^o(i, j)$ refer to the postures of both hands in the first and last key frames of the test sign, respectively right hand key frame 1, right hand key frame 4, left hand key frame 1 and left hand key frame 4. Similarly $P_{R1}^{m_t}(i, j)$, $P_{R4}^{m_t}(i, j)$, $P_{L1}^{m_t}(i, j)$ and $P_{L4}^{m_t}(i, j)$ represent sample t (T samples for each sign) of candidate sign m (M signs) in the selected global feature group. If we show the difference between the test sign and the candidate sign m as $\Delta_o^m d$, we can evaluate differences for all the signs in the selected group, $m = 1, \dots, M$ as given below:

$$\left. \begin{aligned} \Delta_o^m d &= \sum_{t=1}^T \Delta_o^{m_t} d \\ \Delta_o^{m_t} d &= \Delta_o^{m_t} d_{R1} + \Delta_o^{m_t} d_{R4} + \Delta_o^{m_t} d_{L1} + \Delta_o^{m_t} d_{L4} \\ \Delta_o^{m_t} d_{R1} &= \sum_{i=1}^{28} \sum_{j=1}^{28} |P_{R1}^{m_t}(i, j) - P_{R1}^o(i, j)| \\ \Delta_o^{m_t} d_{R4} &= \sum_{i=1}^{28} \sum_{j=1}^{28} |P_{R4}^{m_t}(i, j) - P_{R4}^o(i, j)| \\ \Delta_o^{m_t} d_{L1} &= \sum_{i=1}^{28} \sum_{j=1}^{28} |P_{L1}^{m_t}(i, j) - P_{L1}^o(i, j)| \\ \Delta_o^{m_t} d_{L4} &= \sum_{i=1}^{28} \sum_{j=1}^{28} |P_{L4}^{m_t}(i, j) - P_{L4}^o(i, j)| \end{aligned} \right\} m = 1, \dots, M$$

k nearest neighbor is one of the most common instance-based learning algorithms [15]. k nearest neighbor ranks the differences $\Delta_o^m d$ for $m = 1, \dots, M$, and then uses the labels of k nearest neighbors to predict the meaning of the test sign.

4 Feature Extraction

In this section, we introduce feature extraction steps to obtain the global and the local features.

4.1 Segmentation of Hand and Face Regions

Although many sign language recognition systems [1,3] requires signer to wear some colored gloves, we believe that using the gloves doesn't seem suitable for a system working in public places. For this reason, we benefit of skin tone features to segment the hands and face and use the skin detection method given in [9]. We prefer working with 160x120 pixels resolution in order to make required image processing time lower. Skin regions are tracked by our skin tracker used in [8].

4.2 Motion Extraction and Key Frame Selection

A video sequence consists of consecutive images called frames. Position of a hand in a frame is defined by the center of the rectangle covering the hand region. Since durations of the right and left hand movement during some signs are different [8], we have two distinct trackers for each hand. If the position of the hand changes larger than a threshold, we assign that frame as the start frame. Then, if the position of the hand stays the same, we assign that frame as the end frame.

When a signer tries to perform the same sign twice, different lengths of frame sequences will occur [1]. A Hidden Markov model with skip transitions may handle with it [3]. However, a fixed number of states for all signs is not suitable since Turkish Sign Language contain short signs and long signs. Consequently, we use only 4 frames, called key frames, instead of using all the frames in order not to deal with problems due to the length of the signs. After obtaining the start and end frame of the sign, the whole duration of the sign is divided into 3 equal segments to select the key frames.

5 Experimental Results

Our experimental platform is a personal computer with Intel Pentium IV Mobile 1.9 GHz CPU running Windows XP Professional. We use Philips PCVC 840K CCD camera to capture video. JAVA platform is used to write codes of the learning algorithms (for Hidden Markov model and k nearest neighbor) and the graphical user interface of the system. Our gesture database contains 78 mono-handed and 94 two-handed signs. Each sign is repeated 10 times by a signer. We see that recognizing mono and two handed signs together, in other words, adding the global features of the motionless hand in the feature matrix for a mono-handed sign, does not affect the recognition performance of the system.

2 separate tests are performed. In the first stage test, we train 45 distinct Hidden Markov models corresponding to 45 global feature groups with 1032 (6x172) training samples. For recognition task, the forward algorithm [13] is used to evaluate the probability that a test gesture is generated by the model corresponding to the global feature group i , $P(O|\lambda_i)$ for $i = 1, \dots, 45$. The test sign is determined as a member of the group having the highest probability.

In the second stage test, we suppose that the global feature groups of all the test signs are determined correctly in the first stage to be able to measure the recognition performance of the system based on the local features. k nearest neighbor algorithm for $k = 5$ is operated as explained in Section 3.2 to compare the local features of the test samples and the training samples. The results of both tests are given in Table 1.

As shown in Table 1, there are less test samples in the second stage since there is only one member in 21 global feature groups, which means the local features are not required. Our system determines the global feature groups of 172 gestures with 93.31 % accuracy, and the recognition rate in the local feature processing stage is 91.39 %.

Table 1. The test results of the first and second stages

Stage	First	Second
Test Samples	688 (172x4)	604 (151x4)
Number of Success	642	552
Success Ratio	93.31 %	91.39 %

6 Conclusions and Future Work

In this paper, vision based features, the global features (location and movement of hands) and the local features (hand posture), are used to recognize 172 Turkish Sign Language signs by combining the dynamic and static gesture recognition in two consecutive stages. Sufficient recognition rate has been achieved for person dependent classification. Although the scope of this work is limited to isolated signs, our goal is to design user independent continuous sign language recognition system after completing of our Turkish Sign Language gesture database which will consist of signs performed by 3 hearing impaired people.

References

1. Assan, M., Grobel, K.: Video Based Sign Language Recognition using Hidden Markov Models. Lecture Notes in Computer Science, Vol. 1371. Springer-Verlag, Berlin Heidelberg (1998) 97
2. Fang, G., Gao, W., Zhao, D.: Large Vocabulary Sign Language Recognition Based on Fuzzy Decision Trees. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 34, No.2 (2004)
3. Starnier, T.: Visual Recognition of American Sign Language Using Hidden Markov Models. MS Thesis, Massachusetts Institute of Technology Media Laboratory (1995)
4. Modler, P., Myatt, T.: A Video System for Recognizing Gestures by Artificial Neural Networks for Expressive Musical Control. Gesture Workshop 2003 , Lecture Notes in Computer Science, Vol. 2915. Springer-Verlag, Berlin Heidelberg (2004) 541-548
5. Charayaphan, C., Marble, A.: Image processing system for interpreting motion in American Sign Language. Journal of Biomedical Engineering, Vol. 14 (1992) 419-425
6. Kapuscinski, T., Wysocki, M.: Vision-based recognition of Polish Sign Language. In: Methods in Artificial Intelligence (2003)
7. Sarfraz, M., Yusuf, A. S., Zeeshan, M.: A System for Sign Language Recognition. Using Fuzzy Object Similarity Tracking. Proc. IEEE 9th International Conference on Inf. Visualization (2005)
8. Haberdar, H., Albayrak, S.: Real Time Isolated Turkish Sign Language Recognition from Video Using Hidden Markov Models with Global Features. ISCIS 2005, Lecture Notes in Computer Science, Vol. 3733. Springer-Verlag, Berlin Heidelberg (2005) 677-687

9. Hsu, R. L., Abdel-Mottaleb, M., Jain, A. K.: Face Detection in Color Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, No. 5 Vol. 2. (2002) 696-706
10. Özyürek, A., İlkbaşaran, D., Arık, E.: Turkish Sign Language. Koç University, <http://turkisaretdili.ku.edu.tr> (2004)
11. Turkish Sign Language Manual for Adults. Turkish Ministry of Education, Ankara (1995)
12. Haberdar, H.: Real Time Turkish Sign Language Recognition System From Video Using Hidden Markov Models. MS Thesis, Yıldız Technical University, Computer Science and Engineering Department (2005)
13. Alpaydın, E.: Introduction to Machine Learning. The Massachusetts Institute of Technology Press (2004) 305-326
14. Young, S.: The Hidden Markov Model Toolkit Book Version 3.2.1. Cambridge University Engineering Department. Speech Group and Entropic Research Lab. Inc., Washington DC (2002)
15. Bao, Y., Ishii, N., Du, X.: Combining Multiple k-Nearest Neighbor Classifiers Using Different Distance Functions. International Conference on Intelligent Data Engineering and Automated Learning 2004, Lecture Notes in Computer Science, Vol. 3177. Springer-Verlag, Berlin Heidelberg (2004) 634-641

Speech Emotion Recognition Using Spiking Neural Networks

Cosimo A. Buscicchio, Przemysław Górecki, and Laura Caponetti

Universita degli Studi di Bari, Dipartimento di Informatica
Via E. Orabona 4, 70126, Bari, Italy
{buscicchio, przemyslaw, laura}@di.uniba.it

Abstract. Human social communication depends largely on exchanges of non-verbal signals, including non-lexical expression of emotions in speech. In this work, we propose a biologically plausible methodology for the problem of emotion recognition, based on the extraction of vowel information from an input speech signal and on the classification of extracted information by a spiking neural network. Initially, a speech signal is segmented into vowel parts which are represented with a set of salient features, related to the Mel-frequency cesptrum. Different emotion classes are then recognized by a spiking neural network and classified into five different emotion classes.

1 Introduction

Human social communication depends largely on exchanges of non-verbal signals, including non-lexical expression of emotions in speech. Speech supplies a reach source of information about a speaker's emotional state and the research in the area of emotion recognition is important for developing emotion based human-computer interactions [1]. In fact, recent experiments [2] show that humans use the schemes of interpersonal interaction also when they interact with their computers.

Some early physiological experiments made by Petrushin [3] have shown that humans are able to classify emotions correctly with an average accuracy of 65%. Performance of the artificial emotion recognition systems largely depends on the extraction of relevant features from speech. In the field of signal analysis, many traditional and recent works are based on the analysis of "prosodic" information, which includes pitch, duration and intensity of utterance [4]. For example, Chiu [5] extracted a number of features from speech and a multilayered neural network was employed for the classification. Dellaert [6] compared different classification algorithms and feature selection methods. They achieved 79.5% accuracy with four categories of emotions and five speakers uttering 50 short sentences per category.

In this work, we propose a biologically plausible methodology for the problem of emotion recognition, based on the extraction of vowel information from an input speech signal and on a spiking neural network classifier. Since the parts of

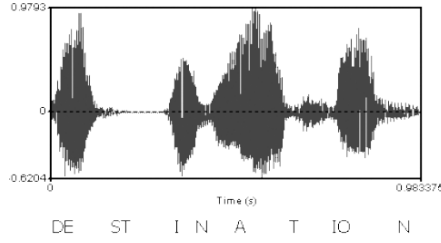


Fig. 1. Sound waveform of the word "destination" for an "angry" stress condition utterance from SUSAS dataset

the speech signal containing consonants do not carry information about emotion, vowel parts are extracted from the speech samples and are represented with a set of salient features, related to the Mel-frequency cepstrum. Different emotion classes are recognized by a spiking neural classifier working on the previously extracted features. The effectiveness of our approach is demonstrated during the experimental session performed with samples selected from the SUSAS database.

The paper is organized as following. In section 2 the feature extraction process is presented. Section 3 describes the details of the spiking neural network classifier. Section 4 presents the network training and classification process. Sections 5 and 6 contain details of the experimental session and the conclusion.

2 Feature Extraction

This section presents our strategy for feature extraction from sentences of spoken words. It is based on the assumption that the sentences are already segmented into separate words. Since the consonants of the word do not carry information about emotions [3], we consider only the vowels. In order to extract vowels from a sentence, Brandt's Generalized Likelihood Ratio (GLR) method is used [7], based on the detection of signal discontinuities. In this way, each sentence is represented as a set of N vowel segments. As an example, the waveform of a word "destination" is shown in figure 1. Note that the vowels correspond to the louder regions of the waveform.

Successively, each segmented vowel is analyzed in order to extract a set of salient features that preserves most of the information. In this work we have considered the Mel-frequency cepstral coefficients [8] as a representation of the vowel. Cepstrum analysis measures the periodicity of the frequency spectrum of a signal that provides information about rate changes in different spectrum bands. The Mel-frequency cepstral coefficients represent cepstrum information in frequency bands, positioned logarithmically on the Mel scale. This scale corresponds to the particular range of pitches that were judged by listeners to be equidistant one from the other. The Mel scale is given by the following formula:

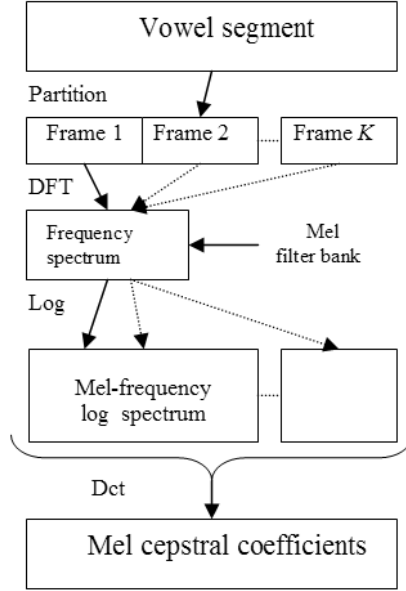


Fig. 2. Vowel feature extraction diagram

$$Mel(f) = 2595 \log \left(1 + \frac{f}{700} \right) ,$$

where f is the frequency value.

Figure 2 shows the overall process of obtaining Mel cepstral coefficients from an input vowel segment. First the signal is divided into small parts called frames having the same time interval. Then, the frequency spectrum of each obtained frame is computed using the Short Fourier Transform. Next, the spectrum is processed by a Mel frequency filter bank, composed of triangular filters spaced uniformly across a logarithmic frequency scale. Finally, Discrete Cosine Transform is applied to Mel-frequency Log Spectrum.

3 Spiking Neural Classifier

Networks of spiking neurons represent a novel class of computational models that raises the level of biological realism by using individual spikes. This allows the incorporation of spatial-temporal information in communication and computation, like real neurons do [9,10]. A spiking neuron generates an action potential, or a spike, when the internal membrane state, called membrane potential, crosses a threshold θ .

There are many different schemes for the use of spike timing information. In this work, we have implemented a network architecture based on an integrate-and-fire neuron model. It approximates the Hodgkin-Huxley model well and captures the generic properties of neural activity [11,12].

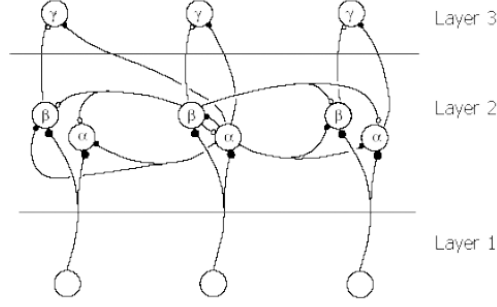


Fig. 3. Spiking neural network classifier. Small dark dots indicate excitatory connections, while small white dots indicate inhibitory connections.

In the framework of the integrate-and-fire model, at time t , the i -th neuron is excited by a spike train $F_j = \{t_j^{(1)}, t_j^{(2)}, \dots, t_j^{(f)}\}$, arriving from one of its predecessors Γ_i , the presynaptic neuron $j \in \Gamma_i$. $t_j^{(f)}$ denotes the time when the neuron j generates a spike f . As a result, the input current I_i of the neuron i is obtained from the following equation, as the sum of all current pulses:

$$I_i(t) = \sum_{j \in \Gamma_i} w_{ij} \sum_f \alpha(t - t_j^{(f)}) ,$$

where the weight w_{ij} describes the synaptic efficacy between the neurons i and j . Moreover, the function α is the spike response that models how the arrival of a single spike charges the membrane potential of the target neuron i . This function, referred to as the postsynaptic current function, can be defined as follows:

$$\alpha(s) = \frac{q}{\tau_s} \exp\left(-\frac{s}{\tau_s}\right) \Theta(s) .$$

The function $\alpha(s)$ is characterized by the amplification factor q and decays exponentially with time constant τ_s . $\Theta(s)$ denotes the step function with $\Theta(s) = 1$ for $s > 0$ and $\Theta(s) = 0$ elsewhere.

Once the input current I_i exceeds a threshold θ , the neuron i discharges and emits a new spike that travels from the body, down the axon, to the next neuron(s). This event is also called depolarization and is followed by a refractory period, during which the neuron is unable to fire. Any neuron generates at most one spike during a simulation interval.

Figure 3 shows the architecture of the spiking neural network employed for classifying patterns into m emotion classes. The network is arranged in three layers without any feedback connection, similar to the one described in [13]:

- Layer 1 is the input layer corresponding to the cortical sensory area. Neurons in layer 1 receive input corresponding to the different frame channels of the input pattern. Each neuron of layer 1 is connected to a small number of neurons in layer 2.

- Layer 2 contains two types of neurons, both of which receive direct excitatory synaptic input from layer 1 afferents: α type cells are excitatory, and β type cells are inhibitory. The neurons of type α and β are connected to a large number of neurons belonging to the same layer. Approximately half of the connections from each neuron are with α neurons, the other half are with β neurons.
- In layer 3, γ neurons receive connections from the α and β neurons in layer 2. The number of connections is defined during the training process. γ neurons are the output units of this system.

4 Network Training and Classification

In order to classify input patterns into m classes by using a spiking neural network, the values of extracted Mel cepstral features must be coded into appropriate spike representation. In our approach, input features are transformed into spike trains, using an average rate coding [14] technique, which assumes that all information carried by a spike train is encoded into the instantaneous average firing rate. This is classically measured by counting spikes in a certain time interval, usually of a fraction of one second.

The network is trained by means of a simple reinforced learning algorithm. For each training pattern belonging to the same emotion class, the pattern is presented to the network and immediately after the firing neurons of layer 2 are observed. The synapses between the firing neurons and the output neuron, corresponding to the correct class, are reinforced, while the others are weakened.

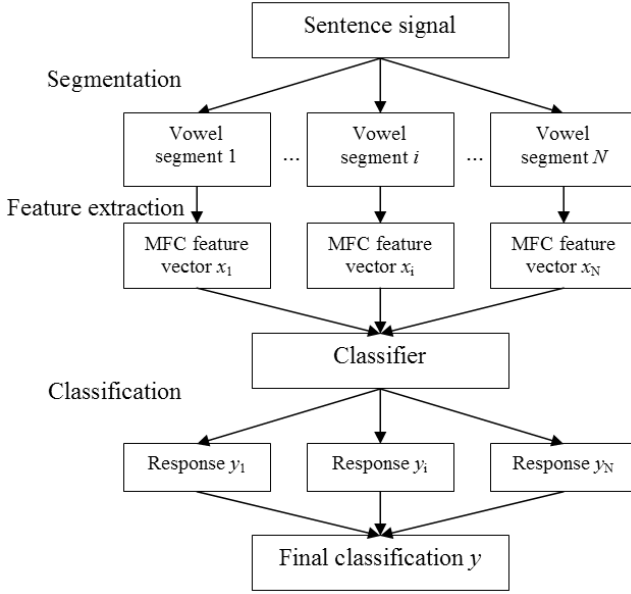
During the classification, each sentence containing N vowels is presented to the network. For each vowel pattern x_i ($i = 1, \dots, N$), the network response is represented as an m -dimensional vector y_i . The value y_i codifies, in the $< 0, 1 >$ confidence interval, the membership degree of the pattern to each of the m emotion classes. The emotion classes and their label vectors are shown in table 1. The final response y to the input sentence is computed as a mean value of all N vowel responses:

$$y = \frac{1}{N} \sum_{i=1}^N y_i .$$

Note that the classifier gives a "soft" classification response. To obtain "hard" classification, the emotion label with the highest response is selected as a final classification result. Fig. 4 depicts the block diagram of the classification process described above.

5 Experimental Session

This section describes the details of the experimental session and presents the final recognition rates of our system, evaluated with the *Speech Under Simulated and Actual Stress* (SUSAS) utterance database [15]. This database consists of

**Fig. 4.** Sentence classification diagram

a vast number of sentences spoken under stress and with a variety of emotions. A total of 32 speakers (13 female, 19 male), of age ranging from 22 to 76 years old, recorded more than 16,000 utterances. Additionally, SUSAS also contains several longer speech files from four Apache helicopter pilots. All sounds are sampled with a frequency of 8 kHz.

During our experiments, we selected speech samples categorized into the five emotion classes described in table 1. There are 1260 utterances in each emotion class: 9 male and 9 female samples of the words listed in table 2; each word is recorded twice.

In order to extract the features necessary for training and testing the neural net, each sample was segmented into vowels. Successively, from each vowel we extracted 24 Mel cepstral coefficients in 0.05s intervals, using Hamming windowed frames of width 0.015s. The feature extraction process was automated using the

Table 1. Emotion classes

Label	Description	Coding
Angry	Simulated anger	[10000]
Clear	Clearly enunciated speech	[01000]
Cond70	Computer workload, high task stress	[00100]
Loud	Loudly spoken speech	[00010]
Soft	Soft or whispered speech	[00001]

Table 2. Susas database words used in the experimental session

SUSAS Vocabulary Set					
Brake	Eighty	Go	Nav	Six	Thirty
Change	Enter	Hello	No	South	Three
Degree	Fifty	Help	Oh	Stand	Wide
Destination	Fix	Histogram	On	Steer	White
East	Freeze	Hot	Out	Strafe	Zero
Eight	Gain	Mark	Point	Ten	

Table 3. Confusion matrix for emotion recognition obtained during experimental session [%]

True label	Response				
	Angry	Clear	Cond70	Loud	Soft
Angry	69.84	12.8	1.86	13.82	1.68
Clear	9.76	74.77	3.23	2.57	9.67
Cond70	9.02	11.03	64.08	10.92	4.96
Loud	15.24	9.84	7.65	63.57	3.71
Soft	0.09	0.21	17.74	4.63	77.34

Table 4. Average classification performance comparison between different models [%]

Emotion class	Yacoub [17]	Kwon [18]	Our approach
Clear	37	93.4	74.77
Angry	70	67.2	69.84
Loud	n/a	48	63.57
Soft	n/a	n/a	77.34

PRAAT software by P. Boersma and D. Weenink [16]. As regards the classifier architecture, we chose the following topology of a spiking neural network:

1. The number of neurons in layer 1 is 24, according to the number of input features.
2. Layer 2 contains 100 type α and 100 type β neurons.
3. In layer 3, the number of γ neurons is equal to 5 classes to be recognized.

The quantitative performance of a spiking neural network was evaluated by considering 20 different splits of the whole data set into a training set (70% of randomly extracted samples) and test set (remaining 30%). The average classification accuracies from all trials that we performed during the session are presented in table 3.

Comparing the results to those of the other authors is difficult, because different datasets and different emotion classes were used in the works of the other authors. In table 4, we present the comparison with other approaches that use the same SUSAS dataset, but different samples. Moreover, our results are promising and are comparable with those obtained by Petrushin [3] during his physiological experiments.

6 Conclusion

In this paper, we have considered the problem of emotion recognition in audio speech signals. We have proposed an approach based on the extraction of Mel cepstrum features from the speech and classification of the features by a spiking neural network classifier. Our system was tested for the classification of five emotion classes, obtaining an average accuracy of 72%. Further work is in progress and the main objective is to improve the classification rates. Different pre-processing techniques and network topologies are being taken into consideration in order to extract the most adequate features and classify them effectively.

References

1. McCauley, L., Gholson, B., Hu, X., Graesser, A.: Delivering smooth tutorial dialogue using a talking head. In Proc. Of wecc-98, Workshop on Embodied Conversational Characters, Tahoe City, California, 1998. AAAI, ACM/SIGCHI.
2. Reeves, B., Nass, C.: The Media Equation. Cambridge University Press, 1996.
3. Petrushin, V. A.: Emotion in speech: Recognition and Application to call centers. Accenture 3773 Willow Rd. Northbrook, IL 60062 - Proceedings of the 1999 Conference on Artificial Neural Networks in Engineering (ANNIE '99)- ASME Press.
4. Sagisaka, Y., Campbell, N., Higuchi, N.: Computing Prosody. Springer-Verlag, New York, NY, 1997.
5. Chiu, C.C., Chang, Y.L., Lai, Y.J.: The analysis and recognition of human vocal emotions. In Proc. International Computer Symposium, pp. 83–88, 1994.
6. Dellaert, F., Polzin, T., Waibel, A.: Recognizing emotion in speech. In Proc. International Conf. on Spoken Language Processing, pp. 1970–1973, 1996.
7. Von Brandt, A.: Detecting and estimating parameters jumps using ladder algorithms and likelihood ratio test. In Proc. ICASSP, Boston, MA, 1983, p. 1017-1020.
8. Rabiner, J.: Fundamentals of speech recognition. Prentice-Hall, 1993.
9. Ferster, D., Spruston, N.: Cracking the neural code. *Science* **(270)** (1995) 756–757.
10. Horn, D., Opher, I.: Collective Excitation Phenomena and Their Applications. In Maass, W. & Bishop, C.M. (eds.) *Pulsed Neural Networks*, MIT-Press (1999).
11. Gerstner, W.: Spiking Neurons. In Maass, W. & Bishop, C.M. (eds.) *Pulsed Neural Networks*, MIT-Press (1999).
12. Gerstner, W., Kempter, R., Leo van Hammen, J., Wagner, H.: Hebbian Learning of Pulse Timing in the Brain Owl Auditory Mass. In Maass, W. & Bishop, C.M. (eds.) *Pulsed Neural Networks*, MIT-Press (1999).
13. Hopfield, J., Brody, C. D.: What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration. *PNAS* **98(3)** (2001) 1282–1287.
14. Maass, W.: Computation with spiking neurons. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press (Cambridge), 2nd edition, 2001.
15. Steeneken, H., Hansen, J.: Speech Under Stress Conditions: Overview of the Effect of Speech Production and on System Performance. *IEEE ICASSP-99: Inter. Conf. on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2079–2082, Phoenix, Arizona, March 1999.

16. Praat homepage: <http://www.fon.hum.uva.nl/praat>
17. Yacoub, S., Simske, S., Lin, X., Burns, J.: Recognition of emotions in interactive voice response systems. In Proc. Eurospeech, Geneva, 2003.
18. Kwon, O-W., Chan, K.-L., Hao, J., Lee, T-W.: Emotion Recognition by Speech Signals. Eurospeech 2003, pages 125 - 128, Sept 2003.

Visualizing Transactional Data with Multiple Clusterings for Knowledge Discovery

Nicolas Durand¹, Bruno Crémilleux¹, and Einoshin Suzuki²

¹ GREYC CNRS UMR 6072,
University of Caen Basse-Normandie, France

² Department of Informatics, ISEE,
Kyushu University, Fukuoka, Japan

Abstract. Information visualization is gaining importance in data mining and transactional data has long been an important target for data miners. We propose a novel approach for visualizing transactional data using multiple clustering results for knowledge discovery. This scheme necessitates us to relate different clustering results in a comprehensive manner. Thus we have invented a method for attributing colors to clusters of different clustering results based on minimal transversals. The effectiveness of our method VISUMCLUST has been confirmed with experiments using artificial and real-world data sets.

1 Introduction

Visualization is of great importance in data mining as the process of knowledge discovery largely depends on the user, who, as a human being, is said to obtain 80% of information from eyesight [7,11,19]. A lot of data mining methods address transactional data. Such data consist of a set of transactions each of which is represented as a set of items and are ubiquitous in real world especially in commerce. As the proliferation of association rule research shows, transactional data have long been an important target of data mining [2]. Attempts for learning useful knowledge from transactional data are numerous and include probabilistic modeling [6], association rule discovery [15], and itemset approximation [1], to name but a few.

Cadez et al. investigate application of probabilistic clustering to obtain profiles from transactional data [6]. They point out visualization as a promising application of their method and have extended their approach for visualization of web navigation patterns [5]. An independent attempt for visualizing time-series data with probabilistic clustering exists [18].

There are a large number of clustering algorithms in machine learning and this situation comes from the fact that the goodness of a clustering result in general depends on the subjective view of the user [10]. We hence believe that visualization based on multiple clustering results can possibly provide various views and outperform the visualization methods based on a single clustering result. In this manuscript, we tackle this issue for transactional data by resolving several difficulties including attribution of colors to clusters from multiple

clustering results. Our method differs from the meta-clustering and clustering aggregation approaches [12] because the color attribution satisfies specific properties for visualization.

The rest of this manuscript is structured as follows. Section 2 defines the problem of visualizing transactional data and provides a survey of related work. We propose our method VisuMClust in Section 3, demonstrate its effectiveness by experiments in Section 4, and conclude in Section 5.

2 Visualizing Transactional Data with Multiple Clustering Results

2.1 Definition of the Problem

Visualizing transactional data with multiple clustering results can possibly realize multiple-view analysis of the user. Such analysis is expected to be effective for a wide range of users. Here we formalize this problem for knowledge discovery.

Transactional data $\mathcal{T}$ consist of n transactions $t_1, t_2, \dots, t_n$ i.e. $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$ each of which is described as a set of items $\mathcal{I}$. A transaction t is a subset of $\mathcal{I}$ i.e. $t \subset \mathcal{I}$. A clustering algorithm i for transactional data outputs a set $\mathcal{C}_i$ of clusters $c_{i,1}, c_{i,2}, \dots, c_{i,m}$ given $\mathcal{T}$ i.e. $\mathcal{C}_i = \{c_{i,1}, c_{i,2}, \dots, c_{i,m}\}$. We also call $\mathcal{C}_i$ a clustering result of the clustering algorithm i . A clustering algorithm where each transaction belongs to only one cluster is called a *crisp* clustering algorithm, otherwise a *soft* clustering (an overlapping between clusters may appear).

The display result D of an information visualization method is diverse and here we avoid restricting its possibilities. We define that our multi-view clustering problem takes $\mathcal{T}$ and multiple clustering results $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_n$ as input and outputs D . The goodness of D is measured by an evaluation measure defined in Section 3.2 and by evidence that the user can discover useful knowledge.

2.2 Related Work

Methods in information visualization can be classified into pattern visualization such as decision-tree visualizer of MineSet [20] and data visualization, which includes VisuMClust. A data visualization method can display raw data such as raw value in the input but can also display transformed data such as an agglomerated value for several raw values. WebCANVAS [5] employs a clustering algorithm which learns a mixture of first-order Markov models and visualizes clusters as navigation patterns on a Web site using membership probabilities of clusters. PrototypeLines [18] employs a clustering algorithm which learns a mixture of multinomial distributions models and visualizes time-series data using an information-theoretic criterion for allocating colors to clusters. These methods are effective but are limited because they rely on a single clustering result.

Ever since the birth of association rule discovery [2], attempts for obtaining potentially useful patterns from transactional data have been made. Examples of such patterns include large itemsets, free itemsets, and closed itemsets but are prohibitively large in number to be employed in visualization. Recently Afrati

et al. proposed to approximate transactional data with a specified number of itemsets and have suggested their use of visualization [1]. Though we think the idea is interesting, we believe that such visualization has the same deficiency as the approach based on a single clustering result.

A recent work [16] proposes a visualization tool to detect changes between two clustering results produced by SOM for different time periods. This work underlines the necessity to associate the used colors between the two clustering results for the visual analysis of the similarities and differences. Nevertheless, this work is limited to two clustering results contrary to our method.

3 VISUMCLUST: Data Visualization with Multiple Clusterings

This section presents our method VISUMCLUST for visualizing data from multiple clustering results for knowledge discovery.

3.1 Outlines of Our Approach

We explain the outline of VISUMCLUST with a small example. The input to the problem is a transactional data set and multiple clustering results shown in Figure 1. The transactional data set (left part of Figure 1) consists of nine transactions $t_1, t_2, \dots, t_9$ described by the set of items $\mathcal{I} = \{A, B, \dots, J\}$. The right part of Figure 1 shows three clustering results $\mathcal{C}_1, \dots, \mathcal{C}_3$. Each clustering result is composed of several clusters (e.g. three clusters for $\mathcal{C}_1$). For each cluster $c_{i,j}$, the table indicates both the transactions belonging to $c_{i,j}$ and the items describing $c_{i,j}$. For instance, the cluster $c_{1,2}$ contains the transactions t_5, t_6, t_7 and it is described by the items D, E . In the area of transactional clustering, it is common to produce results in term of such associations between transactions and itemsets [17].

Id	Items
1	$A \quad J$
2	ABC
3	ABC
4	ABC
5	DE
6	$DE \quad H$
7	$A \quad DEFGH$
8	$A \quad FG \quad I \quad J$
9	HI

Id	Clusters
$\mathcal{C}_1$	$(A; t_1, t_2, t_3, t_4, t_7, t_8) \quad (DE; t_5, t_6, t_7) \quad (I; t_8, t_9)$
$\mathcal{C}_2$	$(AFG; t_7, t_8) \quad (ABC; t_2, t_3, t_4) \quad (DEH; t_6, t_7) \quad (I; t_8, t_9) \quad (J; t_1, t_8)$
$\mathcal{C}_3$	$(ABC; t_2, t_3, t_4) \quad (DE; t_5, t_6, t_7) \quad (I; t_8, t_9) \quad (J; t_1, t_8)$

Fig. 1. Example of a transactional data set and associated multiple clustering results

VISUMCLUST is a generic approach and *any* transactional clustering method (as well crisp as soft) providing clusters with itemsets can be used. In this paper, we use ECCLAT as a clustering method because it is efficient on large transactional data and it enables natural interpretation as a soft clustering method [8].

Given the input, VISUMCLUST allocates a color to each cluster. This allocation should guarantee a “consistent” view of the transactional data set through multiple clustering results: similar clusters in different clustering results are preferably allocated the same color or two similar colors (the method is depicted in Section 3.2). This strategy highlights differences and similarities between the clustering results. The basic idea is to display each transaction as a set of colored rectangles (i.e., rods) each of which is placed in a column based on several clustering results (Figure 2 shows the display result¹ of VISUMCLUST on our small example). Consequently each row represents a clustering result and is represented as a line of rods which corresponds to the visualisation result of the transactional data. The order of transactions is equivalent to the order in the data set. Each rod is displayed by using the colors of the groups of clusters to which the corresponding transaction belongs (Table 1 details the colors of the groups of clusters). If a transaction belongs to a single group of clusters in a clustering result, then its rod has a single color (e.g. t_1). With soft-clustering, a transaction can belong to several groups of clusters and then it is displayed for each clustering result by using all the colors of the groups of clusters to which the transaction belongs (e.g., the next to the last rod of the lines belongs to two clusters in $\mathcal{C}_1$ and $\mathcal{C}_3$ while three in $\mathcal{C}_2$). We believe that this is a simple and natural way to present the data to the user.

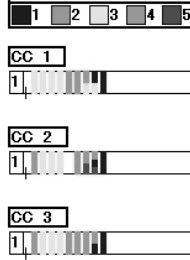


Fig. 2. Example of display result of VISUMCLUST

Table 1. Colors allocated by VISUMCLUST from the clustering results indicated by Figure 1

	1 (blue)	2 (green)	3 (yellow)	4 (orange)	5 (red)
Clusters of $\mathcal{C}_1$	$(I; t_8, t_9)$	$(DE; t_5, t_6, t_7)$	$(A; t_2, t_3, t_4, t_7, t_8)$	$\emptyset$	$\emptyset$
Clusters of $\mathcal{C}_2$	$(I; t_8, t_9)$	$(DEH; t_6, t_7)$	$(ABC; t_2, t_3, t_4)$	$(J; t_1, t_8)$	$(AFG; t_7, t_8)$
Clusters of $\mathcal{C}_3$	$(I; t_8, t_9)$	$(DE; t_5, t_6, t_7)$	$(ABC; t_2, t_3, t_4)$	$(J; t_1, t_8)$	$\emptyset$

This allocation strategy turned out to be effective in interpretation. In Figure 2, we see that the clusters depicted by the color 3 seem reliable. They gather the same transactions (t_2, t_3, t_4) on $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$. The prototypical transactions

¹ All the pictures are available in color at <http://www.info.unicaen.fr/~ndurand/visumclust>

of the clusters of color 2 are t_5 and t_6 and, for the color 1, t_9 . Such transactions may be essential in interpreting these clusters. The transactions t_7 and t_8 seem difficult to allocate in a set of clusters. They may be outliers or their interpretation may be linked to several sets of clusters. Colors 4 and 5 are under represented and the clusters using them may be unreliable. Obviously, these interpretations are based on hypotheses but we argue that they guide the user in knowledge discovery and Section 4 shows the effectiveness of this approach on data sets.

3.2 Color Attribution Based on the Minimal Transversals of a Hypergraph

We start by formalizing the problem of allocation of colors. Let k be the maximal number of clusters for a clustering result. The principle is to select one cluster from each clustering result in order to get the set of the most similar clusters and allocate the same color to these clusters. Then, the selected clusters are removed and the process is iterated until no cluster remains. Finally, each cluster is allocated a color and we call a set of clusters having the same color a *group*.

In this work, we apply the Jaccard coefficient as a similarity measure between clusters because it is very usual but other similarity measures can be used. The Jaccard coefficient is the ratio between the number of common and distinct items between two itemsets (i.e., $Jaccard(I_1, I_2) = \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|}$) [4]. The similarity of a group is related to the notion of intra-cluster similarity used in clustering [4]. It is the average of the similarities of all pairs of clusters (a cluster is also an itemset) of this group. We call the similarity of a group *intra-color similarity* $SimC$ (a color corresponds to a group). A goodness measure S_D of the allocation of colors to the display result D is the average of the intra-color similarities of all groups. The higher S_D is, the better the allocation of colors is.

However, the set of candidate groups is typically huge in number (there are k^n candidate groups) and a naive method which enumerates all groups and computes their similarities fails. Our idea is to set this problem in the hypergraph area [3] so that we can benefit from results in this domain. We will see that it ensures to select at each step a good group with respect to the intra-color similarity.

A hypergraph can be thought as a generalization of a graph because the edges, called hyperedges, can have more than two vertices. A transversal is a set of vertices meeting all hyperedges. A minimal transversal is a transversal with a minimal number of vertices (i.e., T is a minimal transversal if $\forall T' \subset T, T'$ is not a transversal). Finding *all* minimal transversals of a hypergraph is a well-studied problem [9] which has many applications including data mining [13].

The task of color attribution can be represented in terms of a hypergraph $\mathcal{H}$ as follows: the vertices represent the clusters and the hyperedges represent the clustering results. By definition of a transversal, the set of all transversals is the set of all candidate groups (e.g., $T' = \{A, AFG, ABC\}$ is a transversal of $\mathcal{H}_1$ from the running example). Thus the minimal transversals are a subset of the candidate groups. An interesting point of the minimal transversals is that they enable to focus on the set of candidate groups gathering identical clusters through

the clustering results. For instance, $T = \{A, ABC\}$ and T' are two candidate groups but T is better than T' : indeed, with T , the cluster ABC is selected both for $\mathcal{C}_2$ and $\mathcal{C}_3$ (and $Jaccard(ABC, ABC) = 1$ and $SimC(T) = 0.55$) whereas, with T' , AFG is picked for $\mathcal{C}_2$ instead of ABC and $Jaccard(ABC, AFG) = 0.2$ and $SimC(T') = 0.29$ (in other terms, T gathers clusters which are more similar than T'). The set of all minimal transversals is the set of all combinations of clusters through the clustering results gathering the identical clusters and VISUMCLUST uses the minimal transversals as candidate groups. Then, instead of generating and computing the similarities of all the potential candidate groups, only the candidate groups corresponding to the minimal transversals are used. For $\mathcal{H}_1$, the similarities of the seven minimal transversals ($\{I\}$, $\{DE, DEH\}$, $\{DE, ABC\}$, $\{DE, J\}$, $\{DE, AFG\}$, $\{A, ABC\}$, $\{A, J\}$) are computed whereas there are 60 candidate groups (i.e., the product of the cardinalities of the three clusterings results).

Many algorithms aim at finding all minimal transversals of a hypergraph. In our problem, hypergraphs are dense because clustering results have similarities and common clusters. We have chosen the CMT prototype [14] because it is efficient for such hypergraphs due to its pruning criteria. The sketch of the color attribution algorithm is given below.

Input: $\mathcal{C}_1, \dots, \mathcal{C}_n$ where $\mathcal{C}_i = \{c_{i,1}, \dots, c_{i,m}\}$.

Output: $D = (color_1, \dots, color_p)$ where $color_i = \{c_{1,j}, \dots, c_{n,j'}\}$.

Start

0. Color = 0

1. Transform $\mathcal{C}_i$ into a hypergraph $\mathcal{H}$

Repeat step 2-6 until there are no clusters $(c_{i,j})$ without color:

2. Color += 1

3. Compute the minimal transversals of $\mathcal{H}$

4. Select the best candidate $Cand$ (i.e. the higher similarity value)

5. Attribute the current color to each $c_{i,j} \in Cand$

6. Add each $Cand$ to D and remove them from $\mathcal{H}$

End

Specific treatments are performed when few clusters remain to avoid allocating the same color to dissimilar clusters. On the other hand, as a human cannot distinguish too many colors efficiently, VISUMCLUST stops the process after the eighth color has been allocated (if clusters remain, they are colored in black and considered as no relevant).

Table 1 provides the result of the color attribution on the small example. Table 2 indicates the intra-color similarity values for this example.

We now demonstrate the usefulness of our method for the color allocation by comparing it to a baseline one. The baseline method employs a greedy strategy

Table 2. Intra-color similarity values of **Table 3.** Colors allocated by the baseline method from the clustering results indicated by Figure 1

	blue	green	yellow	orange	red
$SimC$	1	0.77	0.55	0.33	0

	blue	green	yellow	orange	red
$\mathcal{C}_1$	A	DE	I	-	-
$\mathcal{C}_2$	AFG	DEH	I	J	ABC
$\mathcal{C}_3$	ABC	DE	I	J	-

to build a group. It starts from the first cluster of C_1 , then selects the cluster of C_2 which is the most similar to the already selected cluster and iterates until C_n . When at least two clusters have been selected, the intra-color similarity is used to pick the next cluster. Then, the process is iterated for the next color. Contrary to our method, the result of the baseline method depends on the order of the clusters. Table 3 shows the result by using the baseline method on the example from Figure 1. Clearly, the choice of *AFG* for the color 1 is inappropriate.

4 Experimental Results

The objective of the experiments is twofold. First, we evaluate the improvement brought by the color attribution method of VISUMCLUST versus the baseline method. Second, we investigate the effectiveness of VISUMCLUST for the knowledge discovery using a real-world geographical data set.

4.1 Data Sets and General Results

We used five well-known benchmarks²: **Mushroom** (8124 transactions, 116 items), **Votes** (435 trans., 32 it.), **Hepatitis** (155 trans., 43 it.), **Ionosphere** (351 trans., 98 it.) and **Titanic** (2201 trans., 8 it.). The geographical real-world database³ (called **Geo** in this paper) addresses 99 items stemmed from demographic and economic indicators about 69 geographical units (41 English counties and 28 French regions). **Geo** is used by several universities to detect the links between these geographical units.

For each dataset, several clustering results were obtained by running ECCLAT with different parameters. Table 4 summarizes the overall characteristics of the results. On **Hepatitis**, **Ionosphere** and **Titanic**, the baseline method creates one additional color. Due to the space limitation, we only provide the display results on **Geo** in Figures 3 and 4.

Table 4. Information about the results

	Mushroom	Votes	Hepatitis	Ionosphere	Titanic	Geo
No. of clusterings	5	3	4	3	4	4
Minimal no. of clusters in a clustering	3	6	3	5	7	3
Maximal no. of clusters in a clustering	5	6	4	6	8	5
No. of colors (VISUMCLUST)	6	7	4	6	8	5
No. of colors (baseline method)	6	7	5	7	9	5

In all cases, the computation time is negligible for both the minimal transversals or the baseline method and the improvement brought by the minimal transversals method does not increase computation time.

Table 5 provides the values of the goodness measure S_D for VISUMCLUST and the baseline method on the data sets. VISUMCLUST clearly outperforms the baseline method in the quality of the color allocation (average gain = 27.1%).

² <http://www.ics.uci.edu/~mlearn/> and <http://www.amstat.org/publications/jse/>

³ <http://atlas-transmanche.certic.unicaen.fr/index.gb.html>

Table 5. Comparison of the average intra-color similarity

	Mushroom	Votes	Hepatitis	Ionosphere	Titanic	Geo
Baseline method	0.554	0.617	0.436	0.342	0.675	0.472
VISUMCLUST	0.658	0.839	0.521	0.389	0.878	0.684
Gain	+18.7%	+35.9%	+19.5%	+13.9%	+30.1%	+44.9%

4.2 Knowledge Discovery on Geographical Real-World Data

We give a qualitative evaluation of VISUMCLUST on **Geo**. The aim is twofold: evaluate the effectiveness of VISUMCLUST in knowledge discovery and compare from a qualitative viewpoint our color attribution method to the baseline method. Figure 3 presents the display results of VISUMCLUST on **Geo** by using our color attribution method and Figure 4 with the baseline method. These figures have been shown to geographers, who are experts of these data.

In Figure 3, the colors 1 (blue) and 2 (green) correspond to English and French units, far from **London** and **Paris**, respectively. These units have common features (e.g., demographic indicators expressing a population getting aged). Among these units, some of them (e.g., **Finistère**, **Morbihan** for France and **Cornwall**, **Devon** for England) are also colored in red in the clustering result $\mathcal{C}_4$ because they have also common demographic characteristics (birth rate, age of the population). On the other hand, the capitals and their surrounding areas (like **Reading** for **London** and **Yvelines** for **Paris**) have the same color (3: yellow). This color has captured dynamic units: large agglomerations with good

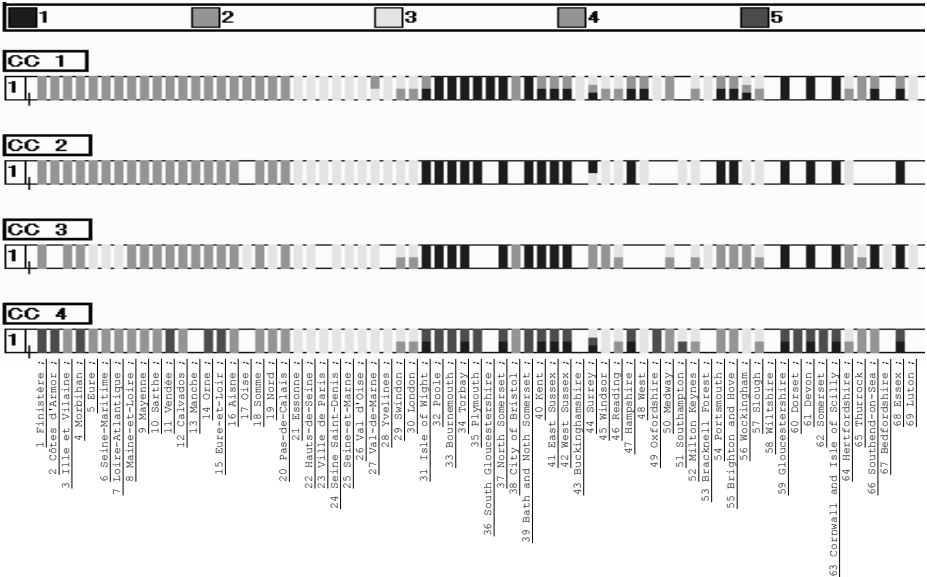


Fig. 3. Results of VISUMCLUST on the geographical database

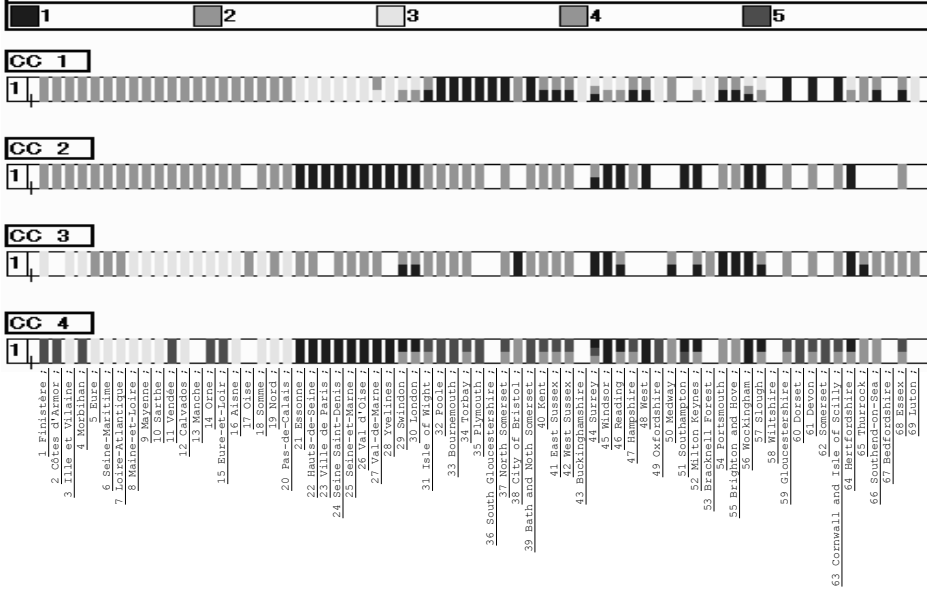


Fig. 4. Results of the baseline method on the geographical database

demographic indicators (low death rate, a lot of young people). The color 4 (orange) only concerns English counties near **London**. The difference with respect to the color 3 lies in economic indicators such as industrial level. These observations have been deduced and validated by the geographers.

On the contrary, in Figure 4, only one color (2: green) gathers geographical units of one country (England). The other colors mix English and French units. It seems difficult to bring out conclusions because transactions are dispersed and many transactions do not have the same color in all the clusterings (contrary to Figure 3). For instance, the color 3 (yellow) concerns **London**, **Paris** and their surrounding areas ($\mathcal{C}_1$) but also some French regions far from Paris (in $\mathcal{C}_3$ and $\mathcal{C}_4$). There is a contradiction because it is proved that large agglomerations and the French regions from west do not have the same demographic and economic features. Similar observations can be done with the color 4 (orange) and some English counties. Compared to Figure 3, only the color 5 gathers the same clusters but it is the last selected color, thus the less reliable.

Figure 4 did not allow to find valid conclusions contrary to Figure 3 and geographers estimate that the main links between English and French units are better captured in Figure 3.

5 Conclusions

In this paper, we have proposed a multi-view visualization method based on multiple clustering results for transactional data. Conventional visualization methods have a deficiency to be used by a wide range of users because the goodness of

a clustering result depends on the user. Our method VISUMCLUST is expected to overcome this problem by providing a consistent view with its color allocation method based on the minimal transversals of a hypergraph. Objective evaluation for the color attribution and subjective evaluation for knowledge discovery are both promising.

An interesting possibility of VISUMCLUST is its capability for comparing different clustering results and choosing the best one. Such an application would require various evaluation measures of clustering results and an effective method for handling user feedback. Other issues include automatic selection of interesting transactions to be visualized and ordering of such transactions.

Acknowledgements. The authors thank Céline Hébert, Frédérique Turbout, Arnaud Soulet and François Rioult for stimulating discussions.

References

1. F. N. Afrati, A. Gionis, and H. Mannila. Approximating a Collection of Frequent Sets. In *Proc. 10th Int. Conf. on Knowledge Discovery and Data Mining (KDD'04)*, pages 12–19, Seattle, WA, August 2004.
2. R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large Database. *ACM SIGMOD*, 22(2):207–216, May 1993. Washington DC, USA.
3. C. Berge. *Hypergraph*. North Holland, Amsterdam, 1989.
4. P. Berkhin. Survey of Clustering Data Mining Techniques. Technical report, Accrue Software, San Jose, CA, 2002.
5. I. V. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Model-Based Clustering and Visualization of Navigation Patterns on a Web Site. *Data Mining and Knowledge Discovery*, 7(4):399–424, 2003.
6. I. V. Cadez, P. Smyth, and H. Mannila. Probabilistic Modeling of Transaction Data with Applications to Profiling, Visualization, and Prediction. In *Proc. 7th Int. Conf. on Knowledge Discovery and Data Mining (KDD'01)*, pages 37–46, San Francisco, California, USA, August 2001.
7. S. K. Card, J.D. Makinlay, and B. Shneiderman, editors. *Readings in Information Visualization*. Morgan Kaufmann, San Francisco, 1999.
8. N. Durand and B. Crémilleux. ECCLAT: a New Approach of Clusters Discovery in Categorical Data. In *Proc. 22nd SGAI Int. Conf. on Knowledge Based Systems and Applied Artificial Intelligence (ES'02)*, pages 177–190, Cambridge, UK, 2002.
9. T. Eiter and G. Gottlob. Identifying the Minimal Transversals of a Hypergraph and Related Problems. *SIAM Journal on Computing Archive*, 24(6):1278–1304, 1995.
10. V. Estivill-Castro. Why So Many Clustering Algorithms - A Position Paper. *ACM SIGKDD Explorations*, 4(1):65–75, June 2002.
11. U. Fayyad, G. G. Grinstein, and A. Wierse. *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, San Francisco, 2002.
12. A. Gionis, H. Mannila, and P. Tsaparas. Clustering Aggregation. In *Proc. 21st Int. Conf. on Data Engineering (ICDE05)*, pages 341–352, Tokyo, Japan, April 2005.
13. D. Gunopulos, R. Khardon, H. Mannila, and H. Toivonen. Data Mining, Hypergraph Transversals, and Machine Learning. In *Proc. 16th Symposium on Principles of Database Systems (PODS'97)*, pages 209–216, Tucson, Arizona, May 1997.

14. C. Hébert. Enumerating the Minimal Transversals of a Hypergraph Using Galois Connections. Technical report, Univ. Caen Basse-Normandie, France, 2005.
15. J. Hipp, H. Güntzer, and G. Nakhaeizadeh. Algorithms for Association Rule Mining - A General Survey and Comparison. *SIGKDD Explorations*, 2(1):58–64, 2000.
16. D. McG. Squire and D. McG. Squire. Visualization of Cluster Changes by Comparing Self-organizing Maps. In *Proc. 9th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'05)*, pages 410–419, Hanoi, Vietnam, May 2005.
17. R. Pensa, C. Robardet, and J-F. Boulicaut. A Bi-clustering Framework for Categorical Data. In *Proc. 9th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'05)*, pages 643–650, Porto, Portugal, October 2005.
18. E. Suzuki, T. Watanabe, H. Yokoi, and K. Takabayashi. Detecting Interesting Exceptions from Medical Test Data with Visual Summarization. In *Proc. 3rd IEEE International Conf. on Data Mining (ICDM'03)*, pages 315–322, 2003.
19. E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2001.
20. C. Westphal and T. Blaxton. *Data Mining Solutions*. John Wiley and Sons, New York, 2000.

An Optimization Model for Visual Cryptography Schemes with Unexpanded Shares

Ching-Sheng Hsu¹, Shu-Fen Tu², and Young-Chang Hou³

¹ Department of Information Management, Ming Chuan University
No. 5, De-Ming Rd., Gui Shan Township, Taoyuan County 333, Taiwan, R.O.C.
cshsu@mcu.edu.tw

² Department of Information Management, Chinese Culture University
No. 55, Huagang Rd., Shihlin District, Taipei City 111, Taiwan, R.O.C.
dsf3@faculty.pccu.edu.tw

³ Department of Information Management, Tamkang University
No. 151, Ying-Chuan Road, Tamshui, Taipei County 251, Taiwan, R.O.C.
ychou@mail.im.tku.edu.tw

Abstract. Visual cryptography schemes encrypt a secret image into n shares so that any qualified set of shares enables one to visually decrypt the hidden secret; whereas any forbidden set of shares cannot leak out any secret information. In the study of visual cryptography, pixel expansion and contrast are two important issues. Since pixel-expansion based methods encode a pixel to many pixels on each share, the size of the share is larger than that of the secret image. Therefore, they result in distortion of shares and consume more storage space. In this paper, we propose a method to reach better contrast without pixel expansion. The concept of probability is used to construct an optimization model for general access structures, and the solution space is searched by genetic algorithms. Experimental result shows that the proposed method can reach better contrast and blackness of black pixels in comparison with Ateniese et al.'s.

1 Introduction

Visual cryptography schemes (VCSs) were first proposed by Naor and Shamir in 1995 [1]. The difference between visual cryptography and traditional ones is the decryption process. Traditional cryptographic methods decrypt secrets by computers; while visual cryptography schemes can decrypt secrets only with human eyes. Therefore, VCSs are practical methods to share secrets when computers are not available. The (k, n) -threshold visual cryptography scheme is a method to encrypt a binary secret image into n shadow images called shares, so that any k or more shares enable the “visual” recovery of the secret image when they are stacked together. However, one cannot gain any secret information by gathering less than k shares. Ateniese et al. [2] extended the (k, n) -threshold access structure to general access structures in the form of $(\Gamma_{\text{Qual}}, \Gamma_{\text{Forb}}, m)$, where Γ_{Qual} denotes a set of qualified sets, Γ_{Forb} denotes a set of forbidden sets, and m is the pixel expansion parameter. Any qualified set $Q \in \Gamma_{\text{Qual}}$ is able to recover the secret image, whereas any forbidden set $F \in \Gamma_{\text{Forb}}$ cannot leak out any secret information. In the study of visual cryptography, pixel expansion and contrast are two primary issues [3]. Since pixel-expansion based

methods encode a pixel to many pixels on each share, the size of the share is larger than that of the secret image [1-6]. Therefore, such schemes not only result in distortion of shares but also consume more storage space.

In this paper, a method using genetic algorithms is proposed to cope with the problems of pixel expansion. Based on the requirements of security and contrast, the basic idea uses the concept of probability to construct an optimization model for general access structures, and the solution space is searched by genetic algorithms. The result shows that, in comparison with Ateniese et al.'s method, the proposed method can reach better contrast and blackness of black pixels in the case with four shares.

2 The Proposed Scheme

2.1 Access Structures

Let $N = \{1, \dots, n\}$ be a set of n shares, and let 2^N denote the set of all subsets of N . Let $\Gamma_{\text{Qual}} \subseteq 2^N$ and $\Gamma_{\text{Forb}} \subseteq 2^N$, where $\Gamma_{\text{Qual}} \cup \Gamma_{\text{Forb}} = \emptyset$. We refer to members of Γ_{Qual} as qualified sets and members of Γ_{Forb} as forbidden sets. Any qualified set $Q \in \Gamma_{\text{Qual}}$ of shares can recover the secret image, whereas any forbidden set $F \in \Gamma_{\text{Forb}}$ of shares cannot leak out any secret information. The pair $(\Gamma_{\text{Qual}}, \Gamma_{\text{Forb}})$ is called the access structure on N . For $A \subseteq 2^N$, we say that A is monotone increasing if for any $B \in A$ and any $C \subseteq N$ such that $B \cup C = \emptyset$, we have $B \cup C \in A$. That is, any superset of the set belonging to A is also in A . We say that A is monotone decreasing if for any $B \in A$ and any $C \subseteq B$ we have that $B \setminus C \in A$. That is, any subset of the set belonging to A is also in A . In the case where Γ_{Qual} is monotone increasing, Γ_{Forb} is monotone decreasing, and $\Gamma_{\text{Qual}} \cup \Gamma_{\text{Forb}} = 2^N$, we say the access structure is strong. In this paper, we assume that access structures are strong.

Example 1: Let $N = \{1, 2, 3, 4\}$ be the set of shares. Suppose that sets $\{1, 4\}$, $\{3, 4\}$, $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, $\{2, 3, 4\}$, and $\{1, 2, 3, 4\}$ can decrypt the secret image. However, other sets must not leak out any secret information. This access structure can be represented as $\Gamma_{\text{Qual}} = \{\{1, 4\}, \{3, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$ and $\Gamma_{\text{Forb}} = 2^N - \Gamma_{\text{Qual}}$.

2.2 Probability and Blackness

From the perspective of digital image halftoning, in a region of a binary image, the higher the density of evenly distributed black pixels is, the darker that region is. Thus, by adjusting the density of evenly distributed black pixels, one can create various degrees of blackness, which range from 0% (pure white) to 100% (pure black). For example, if 70 pixels of a 10×10 image block are randomly selected and colored black, then this block will be interpreted as an area with 70% blackness by human eyes. In other words, if the probability that a pixel is colored black in a binary image block is 0.7, then the blackness of this block will be 70%. Since visual cryptography is characterized by the “visual” decryption process, the secret can be successfully recovered as long as human eyes can distinguish the difference between black and

white regions. Therefore, by controlling the probability of black pixels in an image region, we can realize a visual cryptography scheme without pixel expansion.

To encrypt a secret image without pixel expansion, one can encrypt each secret pixel to a black or a white pixel on each share. Therefore, for n shares, we have 2^n encryption rules to encrypt a secret pixel. Take the $(2, 2)$ -threshold access structure for example; that is, $N = \{1, 2\}$, $\Gamma_{\text{Qual}} = \{\{1, 2\}\}$, and $\Gamma_{\text{Forb}} = \{\{1\}, \{2\}\}$. The four encryption rules for each secret pixel will be “white-white”, “white-black”, “black-white”, and “black-black” at the corresponding positions on the share S1 and the share S2. The corresponding colors of the stacked share (S1 + S2) are “white”, “black”, “black”, and “black”, respectively (see Table 1). The key point is how to determine the probability values of the corresponding encryption rules on the premise that security is assured and that contrast is optimized. To ensure security, we should guarantee that the probability values of the encryption rules for white pixels and that for black pixels should be identical. Suppose that P_w (resp. P_b) denotes the probability that a white (resp. black) pixel is encrypted and stacked as a black pixel on the stacked share of a forbidden set $F \in \Gamma_{\text{Forb}}$. Based on the monotone assumption, it is trivial to prove that perfect secrecy is ensured if P_w equals to P_b for every forbidden set. From the contrast point of view, the secret image can be reconstructed by the stacked share of a qualified set $Q \in \Gamma_{\text{Qual}}$ only if the difference between P_w and P_b is large enough so that human eyes can distinguish between black and white regions.

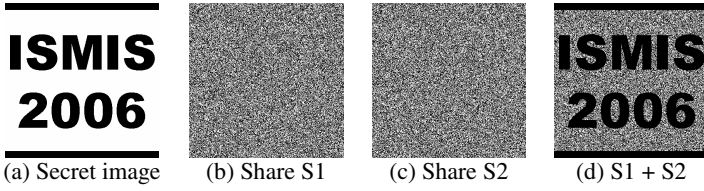
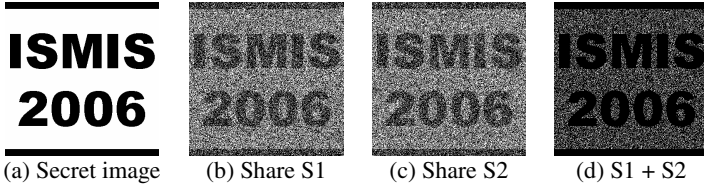
Table 1. Encryption rules of the $(2, 2)$ -threshold VCS

Pixels	ShareS1	Share S2	Stacked share (S1 + S2)	Probability
□	□	□	□	?
	□	■	■	?
	■	□	■	?
	■	■	■	?
■	□	□	□	?
	□	■	■	?
	■	□	■	?
	■	■	■	?

Table 2 shows the probability setting of the encryption rules for the $(2, 2)$ -threshold VCS and their effects on security and contrast. In table 2, (s_{j1}, s_{j2}) denotes the j -th encryption rule on the share S1 and the share S2, and s_{j3} denotes the result of “OR”ing s_{j1} and s_{j2} , where 0 denotes a white pixel and 1 denotes a black pixel. Let c_{0j} (resp. c_{1j}) be the probability that a white (resp. black) pixel is encrypted by the j -th encryption rule. Table 2 shows that, on S1, the probability that a white pixel is encrypted as a black pixel is $FC_{01} = s_{11} \times c_{01} + s_{21} \times c_{02} + s_{31} \times c_{03} + s_{41} \times c_{04} = 0.5$, and the probability that a black pixel is encrypted as a black pixel is also 0.5. Since $FC_{01} = FC_{11} = 0.5$ and $FC_{02} = FC_{12} = 0.5$, security is ensured. On the stacked share (S1 + S2), the probability that a white pixel is encrypted and stacked as a black pixel is $QC_{01} = s_{13} \times c_{01} + s_{23} \times c_{02} + s_{33} \times c_{03} + s_{43} \times c_{04} = 0.5$, and the probability that a black pixel is encrypted and stacked as a black pixel is $QC_{11} = 1$. Since $QC_{11} > QC_{01}$, one can recognize the secret information from (S1 + S2) with eyes (see Fig. 1).

Table 2. An example of the probability setting with good security and good contrast

Pixels	Share S1	Share S2	Stacked share (S1 + S2)	Probability	Probability of being black		
					S1	S2	(S1+S2)
0	$s_{11} = 0$	$s_{12} = 0$	$s_{13} = 0$	$c_{01} = 0.5$	$FC_{01} = 0.5$	$FC_{02} = 0.5$	$QC_{01} = 0.5$
	$s_{21} = 0$	$s_{22} = 1$	$s_{23} = 1$	$c_{02} = 0.0$			
	$s_{31} = 1$	$s_{32} = 0$	$s_{33} = 1$	$c_{03} = 0.0$			
	$s_{41} = 1$	$s_{42} = 1$	$s_{43} = 1$	$c_{04} = 0.5$			
1	$s_{11} = 0$	$s_{12} = 0$	$s_{13} = 0$	$c_{11} = 0.0$	$FC_{11} = 0.5$	$FC_{12} = 0.5$	$QC_{11} = 1.0$
	$s_{21} = 0$	$s_{22} = 1$	$s_{23} = 1$	$c_{12} = 0.5$			
	$s_{31} = 1$	$s_{32} = 0$	$s_{33} = 1$	$c_{13} = 0.5$			
	$s_{41} = 1$	$s_{42} = 1$	$s_{43} = 1$	$c_{14} = 0.0$			

**Fig. 1.** An example of good security and good contrast**Fig. 2.** An example of poor security and moderate contrast

If we change the probability setting of encryption rules in Table 2 to (0.25, 0.25, 0.25, 0.25, 0.00, 0.25, 0.25, 0.50), even though the new probability setting can also make difference between black and white regions on the stacked share (S1 + S2), i.e., $QC_{01} (0.75) < QC_{11} (1.00)$, the probabilities corresponding to white and black pixels on the share S1 and S2 are not identical, i.e., $FC_{01} (0.50) < FC_{11} (0.75)$ and $FC_{02} (0.50) < FC_{12} (0.75)$. Thus, the security this access structure cannot be ensured (see Fig. 2). Our objective is to find a proper probability setting (c_{0j} , c_{1j}) so that the contrast of the stacked share is optimized and the security is ensured as well.

3 The Model for General Access Structures

3.1 Encryption Rule and Probability Matrices

Let $S = [s_{jn}]$ be a $2^n \times n$ Boolean matrix, where $s_{jn} \in \{0, 1\}$. We call S the encryption rule matrix, which represents all of the encryption rules on the set N of n shares. Each

row vector $S_j = [s_{j1}, s_{j2}, \dots, s_{jn}]$ denotes an encryption rule. In the case of $n = 3$, $S_j = [1, 0, 0]$ indicates that a pixel (either black or white) will be encrypted as a black pixel, a white pixel and a white pixel on the first, the second, and the third share, respectively. Let $C = [c_{ij}]$ be a 2×2^n matrix, where $c_{ij} \in [0, 1]$, and

$$\sum_{j=1}^{2^n} c_{ij} = 1 \quad (1)$$

We call C the probability matrix, where c_{0j} (resp. c_{1j}) denotes the probability that a white (resp. black) pixel is encrypted by the j -th encryption rule. Consider the case of $(2, 2)$ -threshold VCS in Table 2, $c_{01} = 0.5$ means that there is 50% of chance that a white pixel ('0') is encrypted by the first encryption rule $[0, 0]$, and $c_{14} = 0.0$ means that there is no chance that a black pixel ('1') is encrypted by the fourth encryption rule $[1, 1]$.

3.2 Security

Let FC_{0k} (resp. FC_{1k}) be the probability that a white (resp. black) pixel is encrypted and stacked as a black pixel on the stacked share of a forbidden set $F_k \in \Gamma_{\text{Forb}}$. If the security is to be assured, the values of FC_{0k} and FC_{1k} must necessarily be the same for each forbidden set F_k . Otherwise, if $FC_{0k} \neq FC_{1k}$, then the variation of frequency may leak out the secret information; thus, security is not ensured. We denote the security index of each forbidden set F_k as follows:

$$\sigma_k = |FC_{1k} - FC_{0k}|. \quad (2)$$

Since we assume that access structures are monotone, we can ensure the security of an access structure $\Gamma = (\Gamma_{\text{Qual}}, \Gamma_{\text{Forb}})$ as long as $\sigma_k = 0$ for every forbidden set F_k . That is, the probability setting of the encryption rules for black pixels and that for white pixels are identical. In Eq.(2), FC_{ik} can be obtained by the following equation:

$$FC_{ik} = C \times \text{OR}(S, F_k) \quad (3)$$

where $\text{OR}(S, F_k)$ denotes the column vector obtained by "OR"ing those columns in S corresponding to the shares of the forbidden set F_k .

3.3 Contrast

Let QC_{0h} (resp. QC_{1h}) be the probability that a white (resp. black) pixel is encrypted and stacked as a black pixel on the stacked share of a qualified set $Q_h \in \Gamma_{\text{Qual}}$. Then, QC_{ih} can be obtained by the following equation:

$$QC_{ih} = C \times \text{OR}(S, Q_h). \quad (4)$$

We define the contrast index of the stacked share of a qualified set Q_h to be

$$\alpha_h = QC_{1h} - QC_{0h}. \quad (5)$$

The larger the value of the contrast index α_h is, the better the contrast of the stacked share of the qualified set Q_h is; that is, the secret information can be recognized by human eyes more easily.

3.4 The Model

We use the form $\Gamma = (\Gamma_{\text{Qual}}, \Gamma_{\text{Forb}})$ to denote an arbitrary access structure with q qualified sets and f forbidden sets for a single secret image. The optimization model for VCSs of $\Gamma = (\Gamma_{\text{Qual}}, \Gamma_{\text{Forb}})$ is formulated as Eq.(6). Since there are 2^n possible encryption rules for each secret pixel, there are totally 2×2^n variables corresponding to all the probability values needed to be solved. Furthermore, each variable is a real number between 0 and 1, and all variables corresponding to the black or white pixels must sum up to be 1. The model also contains f constraint functions with respect to security. Besides, since we need to pursue better contrast of each stacked share of the qualified set $Q \in \Gamma_{\text{Qual}}$, there are totally q objective functions to be optimized.

$$\left. \begin{array}{ll} \max & \alpha_h = QC_{1h} - QC_{0h}, \text{ for } h = 1, 2, \dots, q. \\ \text{s.t.} & \sigma_k = |FC_{1k} - FC_{0k}| = 0, \text{ for } k = 1, 2, \dots, f, \\ & \sum_{j=1}^{2^n} c_{ij} = 1, \text{ for } i = 0, 1, \\ & c_{ij} \in [0, 1], \text{ for } i = 0, 1, \text{ and } j = 1, 2, \dots, 2^n. \end{array} \right\} \quad (6)$$

4 The Application of Genetic Algorithms

Genetic algorithms (GAs), search and optimization procedures, which simulate the natural evolution rules, were proposed by Holland in the 1970s [7]. GAs are composed of a population of chromosomes and several genetic operators. In a population, a chromosome denotes a solution of a specific problem. In addition, GAs use the fitness function to evaluate the goodness of each solution and to guide the direction of search in the solution space. There are three primary operators for GAs: reproduction, crossover and mutation. The reproduction operator is used to choose more survivable chromosomes according to the given fitness function, and then copy these selected chromosomes to the mating pool for further genetic operations. The crossover operator is used to exchange parts of the genes of two chromosomes and then to generate the corresponding offspring. The mutation operator is used sporadically to alter genes randomly; therefore, some important genes that do not present before may appear and some new solutions may be generated from a near converged population. Due to the parallel mechanism for processing a population of chromosomes simultaneously, the power of searching from multiple points makes it very suitable for multi-modal and multi-objective optimization problems.

4.1 Encoding and Decoding

Due to the real number data type of the decision variables in Eq.(6), real parameter encoding method is used; that is, each chromosome is composed of a series of real numbers. The real parameter encoding method is superior in its ability to avoid the

problems of “Hamming cliffs” and loss in precision caused by the binary encoding method [8, 9]; moreover, a real number string is shorter than a binary string. To satisfy the last two constraints in Eq.(6), we pick the $2^n - 1$ real numbers $(x_{i1}, x_{i2}, \dots, x_{i,2^n-1})$ from the range between 0 and 1 as the partition points. Thus, the range between 0 and 1 is divided into 2^n segments. The length of the j -th segment represents the value of c_{ij} . Therefore, the chromosome is encoded as $\mathbf{x} = (x_{01}, x_{02}, \dots, x_{0,2^n-1}, x_{11}, x_{12}, \dots, x_{1,2^n-1})$. After $(x_{i1}, x_{i2}, \dots, x_{i,2^n-1})$ is sorted in an ascending order, the $2^n - 1$ partition points are generated. Let $(x'_{i1}, x'_{i2}, \dots, x'_{i,2^n-1}) = \text{Sort}(x_{i1}, x_{i2}, \dots, x_{i,2^n-1})$, $x'_{i0} = 0$, and $x'_{i,2^n} = 1$, where $i = 0, 1$. Using these partition points, we can decode the chromosomes by $c_{ij} = x'_{ij} - x'_{i,j-1}$.

4.2 Fitness Function

Based on the aforementioned chromosome encoding and decoding schemes, we can formulate the contrast functions as $\theta_h(\mathbf{x}) = \alpha_h$, for $h = 1, 2, \dots, q$. In addition, to deal with the first constraint in Eq.(6), we use the penalty function $\phi(\mathbf{x}) = \sum \sigma_k$, which represents the level of violation of the constraints in the optimization model. Consequently, the fitness functions incorporated with the constraints are written as $\Theta_h(\mathbf{x}) = \theta_h(\mathbf{x}) - \beta \phi(\mathbf{x})$, where β determines the strength of penalty. To solve this multiple objective optimization problem, we use the weighted-sum approach because of its excellences in efficiency and the ease of implementation. The fitness function under the weight-sum approach is formulated as $\mathcal{O}(\mathbf{x}) = \sum \Theta_h(\mathbf{x})$.

4.3 Reproduction, Crossover and Mutation

To deal with negative fitness values, we use the binary tournament selection method. The procedure is as follows: pick two chromosomes randomly from the population, compare their fitness values, and copy the chromosome with higher fitness value to the mating pool. Besides, we employ the simulated binary crossover (SBX) operator [10], which biases solutions near each parent more favorably than solutions away from the parents, to deal with the crossover operation for the real number strings. The advantage of using the SBX operator is that one can control the search power of GAs by controlling the distribution index η_c . Finally, we mutate a specific gene by the random initialization operation. Let x_i be the i -th gene in the real parameter encoded string, let y_i be the result of mutating x_i , and let $x_i^{(U)}$ and $x_i^{(L)}$ be the upper and lower bounds of x_i , respectively. The random initialization operation is formulated as $y_i = r_i(x_i^{(U)} - x_i^{(L)}) + x_i^{(L)}$, where r_i denotes a random number between 0 and 1. Besides, we assume that the probability of mutation for each gene is uniform.

5 Results and Discussions

We deal with an access structures shown in Example 1. The parameters for GAs are listed in Table 3, and the resultant probability matrix C is as follows.

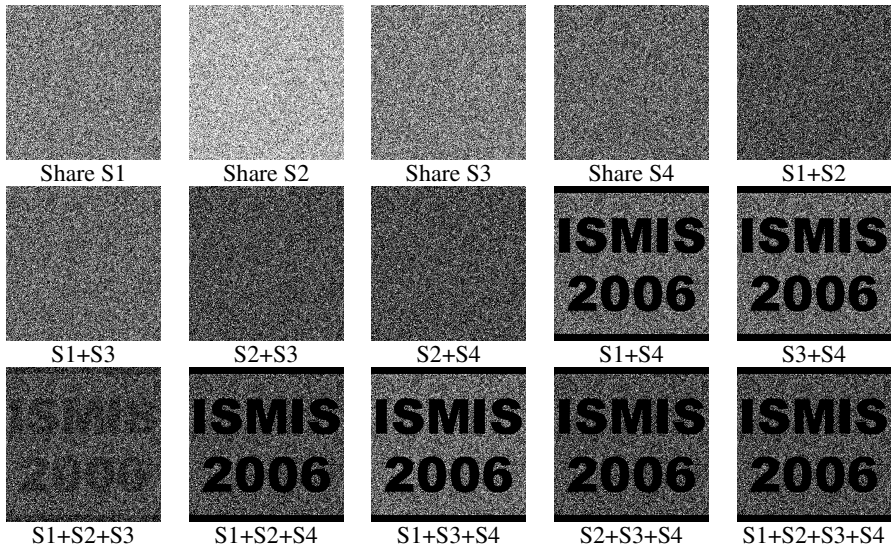
Table 3. Parameters for genetic algorithms

Parameters	Values
Population size	1000
Chromosome length	32
Crossover rate	0.9
Mutation rate	0.01/per gene
Reproduction method	Binary tournament selection
Crossover method	SBX with $\eta_c = 2$
Stop condition	600 generations

$$C = \begin{bmatrix} 0.3 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 & 0.0 & 0.1 & 0.0 & 0.0 & 0.0 & 0.4 & 0.0 & 0.1 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.0 & 0.1 & 0.0 & 0.2 & 0.0 & 0.0 & 0.0 & 0.1 & 0.3 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 \end{bmatrix} \quad (7)$$

Table 4. The security and contrast analysis for Γ

The security index σ_k of the forbidden set F_k		The contrast index α_h of the qualified set Q_h	
$F_1 = \{1\}$	$\sigma_1 = 0$	$Q_1 = \{1, 4\}$	$\alpha_1 = 0.4$ ($QC_{01} = 0.6, QC_{11} = 1.0$)
$F_2 = \{2\}$	$\sigma_2 = 0$	$Q_2 = \{3, 4\}$	$\alpha_2 = 0.4$ ($QC_{02} = 0.6, QC_{12} = 1.0$)
$F_3 = \{3\}$	$\sigma_3 = 0$	$Q_3 = \{1, 2, 3\}$	$\alpha_3 = 0.1$ ($QC_{03} = 0.7, QC_{13} = 0.8$)
$F_4 = \{4\}$	$\sigma_4 = 0$	$Q_4 = \{1, 2, 4\}$	$\alpha_4 = 0.3$ ($QC_{04} = 0.7, QC_{14} = 1.0$)
$F_5 = \{1, 2\}$	$\sigma_5 = 0$	$Q_5 = \{1, 3, 4\}$	$\alpha_5 = 0.4$ ($QC_{05} = 0.6, QC_{15} = 1.0$)
$F_6 = \{1, 3\}$	$\sigma_6 = 0$	$Q_6 = \{2, 3, 4\}$	$\alpha_6 = 0.3$ ($QC_{06} = 0.7, QC_{16} = 1.0$)
$F_7 = \{2, 3\}$	$\sigma_7 = 0$	$Q_7 = \{1, 2, 3, 4\}$	$\alpha_7 = 0.3$ ($QC_{07} = 0.7, QC_{17} = 1.0$)
$F_8 = \{2, 4\}$	$\sigma_8 = 0$		

**Fig. 3.** The experimental result for the access structure Γ

The security and contrast for the resultant probability matrices C are analyzed in Table 4. The shares and stacked results generated according to C are shown in Fig. 3. We can see from Table 4 that the security indices of all the forbidden sets are zero; therefore, C is secure. On the other hand, the contrast indices of the qualified sets $Q_1 \sim Q_7$ are $\alpha_1 = 0.4$, $\alpha_2 = 0.4$, $\alpha_3 = 0.1$, $\alpha_4 = 0.3$, $\alpha_5 = 0.4$, $\alpha_6 = 0.3$, and $\alpha_7 = 0.3$, respectively. Observing Fig. 3, we can say that one can easily recognize the hidden secret from the stacked shares of the qualified sets. Besides, we can also see from Table 4 that the blackness of black regions of the qualified sets Q_1 , Q_2 , and $Q_4 \sim Q_7$ are 100%; therefore, we realized perfect reconstruction of black pixels on the stacked shares of these qualified sets in this experiment.

Ateniese et al. [4] proposed a pixel-expansion-based method to construct basis matrices for the same access structures shown in Example 1, and their basis matrices M_0 for white pixels and M_1 for black pixels with minimum pixel expansion $m^* = 5$ are listed below.

$$M_0 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (8)$$

Using M_0 and M_1 , the contrast indices of the qualified sets $Q_1 \sim Q_7$ are 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, and 0.4, respectively. Comparing with our results, we found that the average contrast is 37.7% higher than that proposed by Ateniese et al. In addition, we reached perfect reconstruction of black pixels on six qualified sets. However, in Ateniese et al.'s method, only the qualified set $\{1, 2, 3, 4\}$ has 100% of blackness for black pixels. We can conclude that our method is superior to Ateniese et al.'s in contrast and blackness of black pixels. Moreover, we do not need to expand pixels.

6 Conclusions

Most visual cryptographic methods need to expand pixels so that the size of each share is larger than that of the secret image. Pixel expansion not only results in distortion of the shares but also consumes more storage space. Consequently, it leads to the difficulty in carrying these shares and the requirement of more storage space. This paper proposed a new method without pixel expansion. We used the concept of probability and considered the security and contrast issues to construct an optimization model for general access structures. Then, the solution space is searched by genetic algorithms. In the case of four shares, we found that our method was superior to Ateniese et al.'s in contrast and blackness of black pixels. In the future, we will study the optimization model for sharing multiple secret images among a set of participants.

Acknowledgement

This work was supported in part by a grant from National Science Council of the Republic of China under the project NSC90-2213-E-008-047.

References

1. Naor, M., Shamir, A.: Visual Cryptography, in Advances in Cryptology-EUROCRYPT '94, LNCS 950, Springer-Verlag, (1995) 1-12.
2. Ateniese, G., Blundo, C., De Santis, A., Stinson, D.R.: Visual Cryptography for General Access Structures, Information and Computation, 129(2), (1996) 86-106.
3. Blundo, C., De Santis, A.: Visual Cryptography Schemes with Perfect Reconstruction of Black Pixels , Computer & Graphics, 12(4), (1998) 449-455.
4. Ateniese, G., Blundo, C. De Santis, A., Stinson, D.R.: Constructions and Bounds for Visual Cryptography, in 23rd International Colloquium on Automata, Languages and Programming (ICALP '96), LNCS 1099, (1996) 416-428.
5. Hou, Y.C.: Visual Cryptography for Color Images, Pattern Recognition, Vol. 36, No. 7, (2003) 1619-1629.
6. Hou, Y.C., Tu, S.F.: A Visual Cryptographic Technique for Chromatic Images Using Multi-Pixel Encoding Method, Journal of Research and Practice in Information Technology, Vol. 37, No. 2, (2005) 179-191.
7. Holland, J.H.: Adaptation in Natural and Artificial Systems", Ann Arbor: The University of Michigan Press (1975).
8. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, West Sussex (2001).
9. Srinivas, M., Patnaik, L. M.: Genetic Algorithms: A Survey, Computer, 27(6), (1994) 17-26.
10. Deb, K., Agrawal, R.B.: Simulated Binary Crossover for Continuous Search Space, Complex Systems, 9(2), (1995) 115-148.

A Fast Temporal Texture Synthesis Algorithm Using Segment Genetic Algorithm

Li Wen-hui, Meng Yu, Zhang Zhen-hua, Liu Dong-fei, and Wang Jian-yuan

College of Computer Science and Technology, Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, Jilin University,
Changchun, 130012, P.R. China
myu_jlu@126.com

Abstract. Texture synthesis is a very active research area in computer vision and graphics, and temporal texture synthesis is one subset of it. We present a new temporal texture synthesis algorithm, in which a segment genetic algorithm is introduced into the processes of synthesizing videos. In the algorithm, by analyzing and processing a finite source video clip, Infinite video sequences that are played smoothly in vision can be obtained. Comparing with many temporal texture synthesis algorithms nowadays, this algorithm can get high-quality video results without complicated pre-processing of source video while it improves the efficiency of synthesis. It is analyzed in this paper that how the population size and the Max number of generations influence the speed and quality of synthesis.

1 Introduction

Texture synthesis is of great importance in computer vision and graphics. Many methods on texture synthesis were introduced in the last decade. Heeger & Bergen [1] use color histograms across frequency bands as a texture description. Bonet[2], who first brought non-parameter texture synthesis method, uses a multiresolution filter-based approach to generate textures. Efros & Leung[3] are first to copy pixels directly from input textures to generate new textures. Portilla & Simoncelli[4] use a synthesis procedure by decomposing texture images to complex wavelets and synthesize new images by matching the joint statistics of these wavelets. Wei & Levoy[5] have improved Efros & Leung's method by using a multiresolution image pyramid based on a hierarchical statistical method. Liang [6] and Efros & Freeman[7] copy whole patches from input textures to synthesize new textures.

The work described above mainly deals with static scenes. Recently has been appeared many temporal texture synthesis methods. Arno Schödl creates long videos by rearranging original frames from a short source video[8], and extends video textures to allow for control moving objects in video[9]. Wang&Zhu[10] models the motion of texture particles in source video to synthesize new videos. Doretto[11] uses Auto-Regressive filters to model and edit the complex motion of fluids in video. Kwatra[12] combines volumes of pixels along minimum error seams to create new sequences that are longer than the original video. Bhat[13] analyzes the motion of textured particles in the input video along user-specified flow lines, and synthesizes

seamless video of arbitrary length by enforcing temporal continuity long a second set of user-specified flow lines.

2 SGA-Based Temporal Texture Synthesis

We suggest a temporal texture synthesis method using Genetic Algorithm[14], which is inspired by Video Textures Algorithm[8]. Compared with other algorithms, this method uses an appropriate fitness function in the processes of creating video sequences instead of much complex pre-processing of source video clip. Additionally, constraints of selection operator, crossover operator and mutation operator ensure that this algorithm can get high-quality video results quickly. Besides, we use the Segment Genetic Algorithm(SGA), an improvement for traditional GA, which can reduce the workload of computing fitness when generate the same number of child chromosomes.

2.1 Video Texture

Video texture is a new type of medium, which is in many ways intermediate between a photograph and a video. It provides a continuous, infinitely varying stream of video images[8]. Video texture has three properties as follow.

1. Continuity. Video texture can be played continuously, and there is no cut in it.
2. Variety. Various video textures can be synthesized from a source video clip.
3. Infinity. Video textures are not fixed video clip, but arbitrary-length video contents.

According to the concept and properties of video texture, video texture synthesis can be transformed into a combinatorial optimum problem. Segment Genetic Algorithm can be used to solve it.

2.2 Genetic Algorithm (GA)

GA is based on the biological phenomenon of genetic evolution. The GA maintains a set of solutions known as a population or a generation. GA operates in an iterative manner and evolves a new generation from the current generation by application of genetic operators. The process of basic Genetic Algorithm is as below:

1. Generate random population of n chromosomes (suitable solutions for the problem)
2. Evaluate the fitness $f(x)$ of each chromosome x in the population
3. Create a new population by repeating following steps until the new population is complete
 - 3.1 Select two parent chromosomes from a population according to their fitness. For each chromosome, the better fitness it has, the bigger chance it would be selected;
 - 3.2 Cross over the parents to form new offspring with a crossover probability. If no crossover was performed, offspring is the exact copy of parents.
 - 3.3 Mutate new offspring at each position in chromosome with a mutation probability;
 - 3.4 Place new offspring in the new population

If the end condition is satisfied, stop, and return the best solution in current population; otherwise, go to step 2

2.3 Video Texture Synthesis Using Segment Genetic Algorithm

2.3.1 Similarity Scale and Measure Rule

To synthesize video texture, the first step is to choose a certain similarity scale and to establish an appropriate measure rule to describe the discontinuity of a video sequence. The simplest method to measure the discontinuity is pixel-base method, in which absolute value difference between each pair of frames should be calculated. This type of method is adopted by most existing synthesis algorithms of video texture, however it is sensitive to global motion of video content and local motion of video object. Histogram-based method can overcome the disadvantage. It is considered that, continuous frames have similar global characters in vision, and then differences between histograms of them are less than those between histograms of discontinuous frames. Only global distribution is concerned about histograms, and it is not sensitive to local motion of video object. However, if two discontinuous frames have similar histograms and dissimilar structure, histogram-based method may mistake them for continuous frames. In order to avoid the problem, we adopt an improved histogram-based method[15] to calculate differences between frames. Define n color spaces, set H_{ik} as number of pixels in the k -th space of the i -th frames. Formula to calculate difference between frames is as below.

$$D_{i,j} = \begin{cases} \sum_{k=1}^n \frac{(H_{ik} - H_{jk})^2}{\max(H_{ik}, H_{jk})}, & H_{ik} \neq 0 \text{ or } H_{jk} \neq 0 \\ 0, & H_{ik} = 0 \text{ and } H_{jk} = 0 \\ +\infty, & i = j \end{cases} \quad (1)$$

2.3.2 Using Segment Genetic Algorithm into Temporal Texture Synthesis

Now we apply SGA to synthesize videos:

Encoding. Given a video sequence S with l frames, $S = \{F_1, F_2 \dots F_l\}$, a solution to temporal texture synthesis will be represented as an ordered list of numbers. Define each decimal number $i (i \in [1, l])$ as a gene and let it denote a frame F_i . For instance, (3 4 5 1 2 3 6 7) is the chromosome representation of video sequence $\{F_3, F_4, F_5, F_1, F_2, F_3, F_6, F_7\}$. In the following algorithm description, we assume that there are m chromosomes in the population and each chromosome $C_k (k \in [1, m])$ denotes a solve to temporal texture synthesis. The simple permutation representation is a common and popular representation for solving the order-based problems.

Fitness. In order to make new video sequence continuous in vision, we should compute similarities among frames and cost of transitions. First we should calculate similarity matrix of D . Then we use formula(2) to calculate mathematical expectation E_k and standard error σ_k of transition cost for each chromosome C_k .

$$E_k = \left(\sum_{j=1}^{n-1} D_{c_j, c_{j+1}} + D_{c_n, c_1} \right) \times n^{-1},$$

$$\sigma_k = \sqrt{\left(\sum_{j=1}^{n-1} (D_{c_j, c_{j+1}} - E)^2 + (D_{c_n, c_1} - E)^2 \right) \times n^{-1}}$$
(2)

Here, n is the number of frames in C_k , c_j represents the j -th ($j \in [1, n]$) gene of C_k

For each chromosome C_k if its E_k is small and its σ_k is large, there are less transitions with high transition costs in the video sequence corresponding to C_k . So that there are some obvious abruptions in the video sequence; on the other hand, if its E_k is large and its σ_k is small, there are more transitions with low transition costs in the video sequence corresponding to C_k . The video sequence is seen as repeatedly playing a small clip. It seems that when its E_k and σ_k are both relative small, chromosomes have better continuity and diversity. Therefore, we can define fitness function as a weighted sum of E_k and σ_k . Furthermore, considering that E_k and σ_k may be not at the same quantity-level, we should normalize them at first. Formula of fitness function is as below:

$$Fit(C_k) = \left(\alpha \cdot E_k + \beta \cdot \frac{\sigma_k \cdot \bar{E}}{\bar{\sigma}} \right)^{-1},$$
(3)

Where $\bar{E} = \left(\sum_{t=1}^{m-1} E_t \right) \times m^{-1}$, $\bar{\sigma} = \left(\sum_{t=1}^{m-1} \sigma_t \right) \times m^{-1}$, and $\alpha + \beta = 1$

Where m is the number of chromosomes in the population. α and β are weight coefficients. In our experiments, we obtain the best result videos in vision when $\alpha=0.7$ and $\beta=0.3$.

Segmentation of Chromosome

First, according to its structure and the length L we divide the chromosome C into q chromosomal segments. Each chromosome will randomly create q chromosomal segments $PU_i = \{C_{ij}\}$, here $i=1, 2, \dots, q$ is the number of the segment population,

$j=1,2,...m$ is the number of the chromosomes in the segment population. We also define the fitness of the segment:

$$f^{g+1}(C_{ij}^{g+1}) = Fit(C_{nmj}^{g+1}) = Fit(C_{1,bs}^{g+1} \cup C_{2,bs}^{g+1} \cup ... \cup C_{i-1,bs}^{g+1} \cup C_{ij}^g \cup C_{i+1,bs}^g \cup ... \cup C_{q,bs}^g) \quad (4)$$

Where $g \geq 1$ is the generation; $C_{ram,j}^{g+1}$ is the combination of each chromosome when we find the best chromosome in $g+1$ generation; $C_{i,best}^g$ ($i=1,2,...q$) are the best segments of each segment colony in g generation which are calculated by $f^g(C_{ij}^g)$ ($j=1,2,...m$); similarly, the fitness of each segment in $g+1$ generation is calculated by $f^{g+1}(C_{ij}^{g+1})$; $\cup$ stands for the connection of each segment. When we find the best segment $C_{i,best}^{g+1}$ for the segment population i in the $g+1$ generation, we will use the best segments in generation $g+1$ (when the number of the segment is less than i) and the best segments in generation g (when the number of the segment is more than i) and each segment in segment population i in generation g . We use the formula 4 to calculate the following values:

$$\begin{aligned} & Fit(C_{1,best}^{g+1} \cup C_{2,best}^{g+1} \cup ... \cup C_{i-1,best}^{g+1} \cup C_{i1}^g \cup C_{i+1,best}^g \cup ... \cup C_{q,best}^g) \\ & Fit(C_{1,best}^{g+1} \cup C_{2,best}^{g+1} \cup ... \cup C_{i-1,best}^{g+1} \cup C_{i2}^g \cup C_{i+1,best}^g \cup ... \cup C_{q,best}^g) \dots\dots \\ & Fit(C_{1,best}^{g+1} \cup C_{2,best}^{g+1} \cup ... \cup C_{i-1,best}^{g+1} \cup C_{im}^g \cup C_{i+1,best}^g \cup ... \cup C_{q,best}^g). \end{aligned}$$

the above values are $f^{g+1}(C_{ij}^{g+1})$. The bigger the value of the $f^{g+1}(C_{ij}^{g+1})$ is, the better segment C_{ij}^{g+1} we can get. Finally, we will find the best segment among all m segments, and then name it $C_{i,best}^{g+1}$, which is the best segment among i segments in the $g+1$ generation. During the dividing process, we set the division point outside the continuous frames in order to avoid the degeneration.

Selection Operator. Selection Operator gives preference to better individuals, allowing them to pass on their genes to the next generation. In our experiments, we use roulette wheel method for selection operation. Survivable probability P_s of each individual C is directly determined by its fitness.

$$P_s = \frac{Fit(C)}{\sum_{k=1}^m Fit(C_k)} \quad (5)$$

Here, m is the amount of individuals in population. C_k means the k -th individual in population.

Crossover Operator. This process recombines portions of good existing individuals to create even better individuals. It leads population to move towards the best solution. Single-point crossover is adopted and each time two genes from different individuals are exchanged. Crossover probability is set as a random float in [0.5,1].

Mutation Operator. The purpose of Mutation Operation is to introduce new gene and maintain diversity within the population and inhibit premature convergence. Selection of Mutation probability is much important for GA. If its value is too great, the population will not be to converge, while if too small probability value may lead the population to premature convergence. In this paper, we set the value to 0.05.

During each iteration of GA, the individual having the largest fitness within all individuals in every generation is selected and checked whether its fitness is larger than a given threshold value of convergence V_r . If the fitness is larger or the number of generations is beyond a specified value G_{max} , the algorithm stops iterations and reports the individual having the largest fitness as the best solution.

Process of Searching

- 1) Generate random population of n chromosomes(n video sequences).
- 2) Evaluate the fitness $Fit(C)$ of each chromosome C in the population.
- 3) Set $g=1$;divide each chromosome into q segments, then evaluate the fitness of each segment and named the best segment in each segment population as $C_{i,best}^g$ ($i=1 \dots q$).
- 4) Join each $C_{i,best}^g$ into chromosomes C_{best}^g . If C_{best}^g satisfies the end condition, stop.
- 5) Apply genetic operation on each segment, and generate new child chromosomes.
- 6) $g = g+1$; Calculate $C_{i,best}^g$, then go to 4.

Suppose that each genetic operation will generate l child chromosomes. In traditional GA, we have to compute the fitness l times. In SGA if a segment population generates l child chromosomes, for q segment populations, there will be $lm^{q-1}q$ child chromosomes, and moreover fitness only needs to be computed qm times. However, for traditional GA if there are $lm^{q-1}q$ child chromosomes, we need to compute fitness $lm^{q-1}q$ times. It is obviously that, $lm^{q-1}q$ is more than qm . Therefore, SGA can reduce the workload of computing fitness under the condition of generating the same number of child chromosomes, and then increase the speed of convergence.

3 Results and Analysis

3.1 Results

The current video synthesizing system is implemented in Visual C++6.0. Some video clips designed by the system are shown in the following figures. The system maintains reasonably smooth frames on a PC (Pentium4 2.0GHz). These video clips



Fig. 1. A 25-frame source video of fire, result video synthesized by the algorithm in this paper is fairly convincing as flame jumping



Fig. 2. A 49-frame source video of ocean, our synthesized video preserves continuity and dynamics of seawater in the source clip



Fig. 3. A 45-frame source video of blow-ing grass, grass flapping in our result videos is the best similar to the source clip

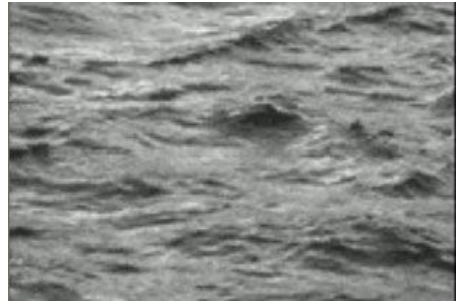


Fig. 4. A 60-frame source video of river, result video synthesized by the algorithm in this paper maintains continuity and dynamics of water in the source clip

demonstrate several different video textures produced by the above-mentioned methods described so far.

3.2 Analysis of Population Size

The population size is crucial to the efficiency and quality of our results. If its value is too small, the algorithm may fail to give us a satisfied solution of temporal texture synthesis; On the other hand, if its value is too large, the algorithm may waste much of calculating time. Therefore, it is of paramount importance to set a proper population size for the efficiency and quality of our algorithm.

In our experiments for each source clip, we set different population sizes to test. When processing the example in Fig.1, we set the number of chromosomes to 30, 50, 100, 150, 300, 500, 800, 1000 and 1500 respectively. Results of our experiments show that the algorithm quickly converges to an unsatisfied solution when the number of chromosomes is tiny; while the number of chromosomes is 300, the algorithm is

able to do a fairly work; as the population size is larger than 300, fitness of the best solution has unobvious change. For the rest figures, satisfactory solutions can be obtained when the population sizes are 500, 500, and 800 respectively.

3.3 Analysis of Max Generation Number

Number of Max generation is another main key to influence the efficiency of our results. When processing the example in Fig.1, we specify that the population size is 300, and set the number of Max generations to 10, 20, 30, 40, 50, 60, 70 and 80 respectively. Results of our experiments show that when the number of Max generations is 20, satisfied solutions can be obtained. For the rest Figures, we specify that the three population sizes are 500, 500 and 800 respectively, satisfied solutions can be obtained when the number of Max generations are 30, 30, and 40.

3.4 Analysis of Computation Complexity

Without computing the part time of pre-processing, the total computation complexity of Video Textures Algorithm is $O(L^2N^2)$, where L is the length of result video sequence and N is the number of transitions. However the total computation complexity of our SGA-based algorithm is $O(PMG)$, where P is the number of segment populations, M is the segment population size, and G is the number of Max generation.

4 Conclusions

In this paper, Segment Genetic Algorithm(SGA) is introduced into the processes of synthesizing videos, and a new temporal texture synthesis algorithm is suggested. This algorithm uses an appropriate fitness function instead of much complex pre-processing of source video clip. Additionally, constraints of selection operator, cross-over operator and mutation operator can ensure this algorithm to get high-quality video results quickly. Compared with other representative temporal texture synthesis algorithms, this algorithm has less computational complexity and improves the efficiency of synthesis.

Although we put more emphasis on the continuities and regularities of video sequences in our algorithm, we need to care for the structures of them. A more appropriate fitness function is anticipated to be found in future research work.

References

1. Heeger D.J. and Bergen, J.R.: Pyramid-based texture analysis/synthesis. Computer Graphics, ACM SIGGRAPH. (1995), 229-238
2. Bonet J.S.D.: Multiresolution sampling procedure for analysis and synthesis of texture images. Computer Graphics, ACM SIGGRAPH. (1997) 361-368
3. Efros A. and Leung T.: Texture synthesis by non-parametric sampling. International Conference on Computer Vision, Vol 2. (1999) 1033-1038

4. Portilla J. and Simoncelli E.P.: A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*. vol.40, No.1. (2000) 49-70
5. Wei L.Y. and Levoy M.: Fast texture synthesis using tree-structured vector quantization. *Computer Graphics, ACM SIGGRAPH*. (2000) 479-488
6. Liang L., Liu C., Xu Y.Q., Guo B. and Shum H.Y.: Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, Vol. 20, No. 3. (2001) 127-150
7. Efros A.A. and Freeman W.T.: Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*. (2001) 341-346.
8. Schödl A., Szeliski R., Salesin D.H. and Essa I.A.: Video textures. *Proceedings of SIGGRAPH 2000*. (2000) 489-498.
9. Schödl A., ESSA I. A. Controlled animation of video sprites. *Proceedings of ACM Symposium on Computer Animation 2002*, (2002) 121~127
10. Wang Y., Zhu S. C. A generative model for textured motion: Analysis and synthesis. In: *Proceedings of European Conf. on Computer Vision (ECCV)2002*, Copenhagen, (2002) 583~598
11. Doretto G., Chiuso A., Soatto S., And Wu Y. Dynamic textures. *International Journal of Computer Vision*2003, (2003) 51(2): 91~109.
12. Kwatra V., Schödl A., Essa I. A., Turk G., and Bobick A. Graphcut Textures: Image and Video Synthesis Using Graph Cuts. In *Proceedings of ACM SIGGRAPH 2003*. (2003) 277~286
13. Bhat K.S., Seitz S.M., Hodgins J.K., and Khosala P.K., Flow-based video synthesis and editing. In *Proc. of ACM SIGGRAPH 2004*. (2004), 360~363
14. Holland, J.: *Adaptation in Natural and Artificial Systems*. University of Michigan Press. (1975)
15. Nagasaka A., Tanaka Y. Automatic video indexing and full-video search for object appearances. In: Knuth E. , Wegner L. M. eds. *IFIP Proceedings of Visual Database Systems*. Amsterdam, The Netherlands: North-Holland, 1992, 113-127

Quantum-Behaved Particle Swarm Optimization with Immune Operator

Jing Liu, Jun Sun, and Wenbo Xu

Center of Intelligent and High Performance Computing,
School of Information Technology, Southern Yangtze University
No. 1800, Lihudadao Road, Wuxi,
214122 Jiangsu, China
{liujing_novem, sunjun_wx, xwb_sytsu}@hotmail.com

Abstract. In the previous paper, we proposed Quantum-behaved Particle Swarm Optimization (QPSO) that outperforms traditional standard Particle Swarm Optimization (SPSO) in search ability as well as less parameter to control. However, although QPSO is a global convergent search method, the intelligence of simulating the ability of human beings is deficient. In this paper, the immune operator based on the vector distance to calculate the density of antibody is introduced into Quantum-behaved Particle Swarm Optimization. The proposed algorithm incorporates the immune mechanism in life sciences and global search method QPSO to improve the intelligence and performance of the algorithm and restrain the degeneration in the process of optimization effectively. The results of typical optimization functions showed that QPSO with immune operator performs better than SPSO and QPSO without immune operator.

1 Introduction

Particle Swarm Optimization (PSO), introduced by Kennedy and Eberhart in 1995, is a population-based evolutionary optimization method [4], inspired by the collective behaviors of birds and flocks, which has been applied in many kinds of engineering problem widely. But in real applications, premature convergence and low intelligence of PSO algorithm have become the main shortcomings. This is due to a decrease of diversity in search space that leads to a total implosion and ultimately fitness stagnation of the swarm. Then many improved algorithms based on PSO algorithm have been proposed by the researchers [2][5][12][13]. These various revised version, generally speaking, can enhance the convergence performance and the search ability of PSO considerably. However, studies by F.van den Bergh demonstrated that original PSO could not guarantee the algorithm to find out the global optimum with probability 1[3]. So keeping to the philosophy of PSO, a global convergence-guaranteed search technique, Quantum-behaved Particle Swarm Optimization was proposed by Jun Sun et al [7][8]. However, the particles in PSO algorithm and Quantum-behaved PSO algorithm are also adjusted according to their own experience to fly toward their best solution, so during the latter period of search the convergence speed is remarkably slow and the search procedure will stagnate when the convergence value reaches a determined precision. Moreover, although QPSO is a global convergent optimization

technique, its operation is usually performed randomly in the sense of probability. So degeneration will occur inevitably in the process of optimization to a certain extent. As we known, the biological immune system composed of tissue, apparatus and cell, is a high complex system, which have many properties including diversity, immune memory, dynamic learning and self-monitoring. Inspired by the theory of immunology, various optimization methods, algorithms and technology were developed [1][10][14]. In this paper, to increase the intelligence of evolutionary algorithm, we proposed Quantum-behaved Particle Swarm Optimization with immune operator algorithm, which incorporates the immune mechanism in life sciences and global search method QPSO to restrain the degeneration in the process of optimization.

The rest of the paper is organized as follows: The standard PSO algorithm and the Quantum-behaved PSO algorithm are explained in section 2. In section 3, the biology immune system is briefly introduced, and then the mechanism of immune is introduced into QPSO algorithm to construct the immune operator, which effectively incorporates QPSO algorithm and immune system. Section 4 shows the experimental settings and comparative results of QPSO with immune operator followed by conclusions in section 5.

2 PSO and Quantum-Behaved PSO

2.1 Dynamics of Standard PSO

In a classical PSO system proposed by Kennedy and Eberhart [4], each particle flies in a D-dimensional space S according to its own historical experience and others. The velocity and location for the i th particle is represented as $\vec{v}_i = (v_{i1}, \dots, v_{id}, \dots, v_{iD})$ and $\vec{x}_i = (x_{i1}, \dots, x_{id}, \dots, x_{iD})$, respectively. The particles are manipulated according to the following equation:

$$x_{id} = x_{id} + v_{id} \quad (1a)$$

$$v_{id} = w \cdot v_{id} + c_1 \cdot \text{rand}() \cdot (p_{id} - x_{id}) + c_2 \cdot \text{rand}() \cdot (p_{gd} - x_{id}) \quad (1b)$$

where c_1 and c_2 are acceleration constants, $\text{rand}()$ are random values between 0 and 1.

In (1a), the vector P_i is the best position (the position giving the best fitness value) of the particle i , vector P^g is the position of the best particle among all the particles in the population. Parameter w is the inertia weight to balance the ability of exploration and exploitation [15], which does not appear in the original PSO, as denoted Standard Particle Swarm Optimization (SPSO). In [11], M. Clerc and J. Kennedy analyze the trajectory and prove that, whichever model is employed in the SPSO algorithm, each particle in the PSO system converges to its local point P , whose coordinates are $p_d = (\varphi_{1d} p_{1d} + \varphi_{2d} p_{gd}) / (\varphi_{1d} + \varphi_{2d})$ so that the best previous position of all particles will converge to an exclusive global position with $t \rightarrow \infty$, where $\varphi_{1d}, \varphi_{2d}$ are random numbers distributed uniformly on $[0, 1]$.

2.2 Dynamics of Quantum PSO

In the quantum model of a PSO, the state of a particle is depicted by wave function $\Psi(\vec{x}, t)$, instead of position and velocity. The dynamic behavior of the particle is widely divergent from that of the particle in traditional PSO systems in that the exact values of x and v cannot be determined simultaneously. We can only learn the probability of the particle's appearing in position x from probability density function $|\Psi(X, t)|^2$, the form of which depends on the potential field the particle lies in.

The particles move according to the following iterative equation [7][8]:

$$x(t+1) = p \pm \beta * |mbest - x(t)| * \ln(1/u) \quad (2a)$$

Where

$$mbest = \frac{1}{M} \sum_{i=1}^M P_i = \left(\frac{1}{M} \sum_{i=1}^M P_{i1}, \frac{1}{M} \sum_{i=1}^M P_{i2}, \dots, \frac{1}{M} \sum_{i=1}^M P_{id} \right) \quad (2b)$$

$$P_{id} = \varphi * p_{id} + (1 - \varphi) * p_{gd}, \varphi = rand() \quad (2c)$$

$mbest$ (Mean Best Position) is defined as the mean value of all particles' the best position, φ and u are random number distributed uniformly on $[0,1]$, respectively. β , Contraction-Expansion Coefficient, is the only parameter in QPSO algorithm.

3 Biological Immune System and Immune Operator

3.1 Biological Immune System

The biological immune system is highly complicated and appears to be precisely tuned to the problem of detecting and eliminating infections. The immune system is compelling because it exhibits many properties that we would like to incorporate into artificial systems: it is diverse, distributed, error tolerant, dynamic and adaptable. These properties endow the immune system certain key characteristic: diversity to generate lots of antibodies against all kinds of antigens and to improve robustness on both a population and individual level; immune regulate to restrain or stimulate some antibodies; immune memory of those infections to recognize previously encountered antigens and facilitate future response.

3.2 Immune Operator

In QPSO algorithm, the particles with high fitness are always expected to retain during the process of particle updating. However, once such particles are excessively centralized, QPSO algorithm is much easy to trap in the local optimum and lose those particles provided with low fitness but a preferable evolutionary tendency. On the other hand, the immune algorithm [9] calculated by information entropy to attain the density of antibody has some shortcomings, such as complex calculation, parameters setting limited to experience and slow convergence speed due to redundancy computation information. So the immune operator based on the vector distance to calculate

the density of antibody is applied in the proposed algorithm---Quantum-behaved PSO with immune operator. To be more exact, the idea of constructing the immune operator is to perform immune memory, immune selection and immune regulation, which prevent the degeneracy in the process of search.

According to the mechanism of immune system, each B-cell produces identical antibodies specific to only one antigen. In our proposed algorithm, antigen, B-cell and antibody correspond to the objective function of optimization problem, optimum x_i and the fitness function $f(x_i)$ respectively. And N antibodies are composed of a non-empty immune system Set X . The distance of antibody $f(x_i)$ in Set X can be defined as:

$$\rho(x_i) = \sum_{j=1}^N |f(x_i) - f(x_j)| \quad (3)$$

And the density of antibody is evaluated by

$$Density(x_i) = \frac{1}{\rho(x_i)} = \frac{1}{\sum_{j=1}^N |f(x_i) - f(x_j)|} \quad (4)$$

Thus the probability function based on density of antibody can be given by

$$P_s(x_i) = \frac{\rho(x_i)}{\sum_{i=1}^N \rho(x_i)} = \frac{\sum_{j=1}^N |f(x_i) - f(x_j)|}{\sum_{i=1}^N \sum_{j=1}^N |f(x_i) - f(x_j)|} \quad (5)$$

From the Equation (5), we can see the more antibody similar to antibody i , the less probability the antibody i is selected to attain, whereas the less antibody similar to antibody i , the greater probability to select. Thus not only the particles with high fitness are expected to retain, but also the particles with low fitness value but a preferable evolutionary tendency seem to be a certain probability of continuing to search in the search space. Therefore, the diversity of antibody and much more intelligence of the algorithm are guaranteed by the selection mechanism based on the regulation of the antibody thickness in immune system.

3.3 QPSO- Immune Algorithm Pseudocode

The Quantum-behaved PSO algorithm with immune operator is described as follows:

Initialize M particles

IM= p_g ;

Do

β linearly decreases from 1.0 to 0.5 with generation;

 find out the mbest of the swarm;

 if PM<rand(0,1)

 mbest=Cauchy(mbest) [6];

 update new particles according to evolutionary equation Eq(2a) of QPSO;

```

generate randomly other new N particles;
perform immune selection among (M+N)particles accord-
    ing to probability function Equation(5);
replace particles with lowest value with IM;
if f_pbest(g) < MINIMUM
    MINIMUM = f_pbest(g)
    IM =  $p_g$ ;

```

Until the termination criterion is met

The parameter IM is to retain immune memory, which records the best position p_g among all the particles, where g is the index of the best particle p_g . PM denotes a fixed mutation probability varying with optimization functions, which is applied in QPSO algorithm with mutation operator [6], that is the Quantum-behaved PSO with immune operator is based on the Quantum-behaved PSO with mutation operator algorithm. And the number of particles to replace with IM varied with objective function.

It is also noted that in view of diversity, the mechanism is not applied the method of roulette but to select among the total new updating particles and randomly generated particles to increase the diversity of population.

4 Experimental Setting and Results

The only parameter β decreases linearly from 1.0 to 0.5 with generations in the proposed algorithm.

To test the performance of QPSO with immune operator, the optimization functions commonly used in the evolutionary computation literature will be demonstrated. Table 1 gives the test functions: Rastrigrin, Rosenbrock and Griewank, mathematics expression, its initialization range and the corresponding limits to the search space and its function value.

Table 1. Test Functions and parameter configuration

F	Mathematic expression	Initial Range	$f_{\min}$	$X_{\max}$
F_1	$f(x) = \sum_{i=1}^n (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	(15,30)	0	100
F_2	$f(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	(2.56, 5.12)	0	10
F_3	$f(x) = \frac{1}{4000} \sum_{i=1}^n (x_i - 100)^2$ $-\prod_{i=1}^n \cos(\frac{(x_i - 100)}{\sqrt{i}}) + 1$	(300,600)	0	600

As in [7], for each function, three different dimension sizes, 10,20 and 30 are tested. The corresponding maximum generations are 1000, 1500 and 2000 respectively. And the population size is set to 20, 40 and 80.

Table 2. Experimental Values For Rosenbrock Function

P	D	G	SPSO		QPSO		QPSO-IM	
			Mean	St. Dev	Mean	St. Dev	Mean	St. Dev
20	10	1000	94.1276	194.3648	59.4764	153.0842	6.7040	0.7417
	20	1500	204.336	293.4544	110.664	149.5483	17.0532	0.9449
	30	2000	313.734	547.2635	147.609	210.3262	27.2594	1.0407
40	10	1000	71.0239	174.1108	10.4238	14.4799	7.6098	0.7624
	20	1500	179.291	377.4305	46.5957	39.5360	18.9514	0.1464
	30	2000	289.593	478.6273	59.0291	63.4940	29	0
80	10	1000	37.3747	57.4734	8.63638	16.6746	6.4344	0.5601
	20	1500	83.6931	137.2637	35.8947	36.4702	18.3756	0.5236
	30	2000	202.672	289.9728	51.5479	40.8490	28.9359	0.1911

Table 3. Experimental Values For Rastrigrin Function

P	D	G	SPSO		QPSO		QPSO-IM	
			Mean	St.Dev	Mean	St. Dev	Mean	St.Dev
20	10	1000	5.5382	3.0477	5.2543	2.8952	0	0
	20	1500	23.1544	10.4739	16.2673	5.9771	0	0
	30	2000	47.4168	17.1595	31.4576	7.6882	0	0
40	10	1000	3.5778	2.1384	3.5685	2.0678	0	0
	20	1500	16.4337	5.4811	11.1351	3.6046	0	0
	30	2000	37.2796	14.2838	22.9594	7.2455	0	0
80	10	1000	2.5646	1.5728	2.1245	1.1772	0	0
	20	1500	13.3826	8.5137	10.2759	6.6244	0	0
	30	2000	28.6293	10.3431	16.7768	4.4858	0	0

Table 4. Experimental Values For Griewank Function

P	D	G	SPSO		QPSO		QPSO-IM	
			Mean	St. Dev	Mean	St. Dev	Mean	St.Dev
20	10	1000	0.09217	0.08330	0.08331	0.06805	0.0724	0.0432
	20	1500	0.03002	0.03225	0.02033	0.02257	0.0142	0.0189
	30	2000	0.01811	0.02477	0.01119	0.01462	0.0125	0.0174
40	10	1000	0.08496	0.07260	0.06912	0.05093	0.0746	0.0720
	20	1500	0.02719	0.02517	0.01666	0.01755	0.0183	0.0179
	30	2000	0.01267	0.01479	0.01161	0.01246	0.0081	0.0091
80	10	1000	0.07484	0.07107	0.03508	0.02086	0.0977	0.0786
	20	1500	0.02854	0.02680	0.01460	0.01279	0.0101	0.0092
	30	2000	0.01258	0.01396	0.01136	0.01139	0.0065	0.0097

Table 2-Table 4 show the mean fitness value of the best point found by the end of the trial for the 50 trials along with the standard deviation for each set of trials in the experiment. The tests showed the mean best point found by QPSO-immune to be statistically significantly better for all functions except f_3 . For function f_3 , the improvements are not so remarkable but its theory optimum value 0 can be obtained many times among 50 trials. And especially for f_2 , the results obtained from the experiments are equal to its theory value zero for all population and dimension.

5 Conclusions

In this paper, the immune operator based on the regulation of the antibody thickness in immune system is introduced into the Quantum-behaved PSO algorithm to increase intelligence for simulating the ability of human beings to deal with problems. The performance of the QPSO algorithm with immune operator has been extensively investigated by experimental studies of three functions well studied in the literature. The experimental results show that QPSO with immune operator improves the convergence precision and provides an extend improvement on global search ability. Further research would investigate and incorporate the mechanism of immune system, such as vaccination and immune memory into QPSO algorithm to increase its intelligence of search.

References

1. D.Dasgupta.: Artificial neural networks and artificial immune systems: similarities and differences. IEEE International Conference on Systems, Man and Cybernetics (1997) 873-878
2. F.van den Bergh, A.P.Engelbrecht: A new locally convergent particle swarm optimizer. IEEE International Conference on systems, Man and Cybernetics (2002)
3. F.van den Bergh.: An analysis of Particle swarm optimizers. Phd Thesis, University of Pretoria (2001)
4. J. Kennedy and R. Eberhart.: Particle Swarm Optimization. Proceedings of IEEE Int. Conf. on Neural Network (1995) 1942-1948
5. J.Kennedy.: Small worlds and mega-minds: effects of neighbourhood topology on particle swarm performance. Proc. Congress on Evolutionary Computation (1999) 1931-1938
6. Jing Liu, Jun Sun and W. Xu.: Quantum-behaved Particle Swarm Optimization with mutation operator. IEEE Tools with Artificial Intelligence (2005) 237-240
7. J. Sun, B. Feng and W. Xu.: Particle Swarm Optimization with Particles Having Quantum Behavior. IEEE Proc. of Congress on Evolutionary Computation (2004) 325-331
8. J. Sun *et al.*: A Global Search Strategy of Quantum-behaved Particle Swarm Optimization. IEEE conference on Cybernetics and Intelligent Systems (2004) 111-116
9. JS Chun, MK Kim, HK Jung, SK Hong.: Shape optimization of electromagnetic devices using immune algorithm. IEEE Trans on Magnetics, vol33, no2 (1997) 1876-1879
10. L Jiao, L Wang.: A novel genetic algorithm based on immunity. IEEE Trans on Systems, Man and Cybernetics, vol 30,no 5 (2000) 552-561
11. M. Clerc and J. Kennedy.: The Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space. IEEE Transaction on Evolutionary Computation, vol.6 (2002) 58-73
12. P. J. Angeline.: Using selection to improve particle swarm optimization. Proceedings of the IEEE Conference on Evolutionary Computation, ICEC (1998) 84-89
13. P.N.Suganthan.: Particle swarm optimizer with neighborhood operator. Proc Congress on Evolutionary Computation (1999) 1958-1962
14. Steven A.Frank.: The design of natural and artificial adaptive systems. Academic Press, New York, M.R. Rose and G.V. Lauder edition (1996)
15. Y. Shi and R. Eberhart.: Empirical study of particle swarm optimization. Proc. Congress on Evolutionary Computation (1999) 1945-1950

Particle Swarm Optimization-Based SVM for Incipient Fault Classification of Power Transformers

Tsair-Fwu Lee^{1,2}, Ming-Yuan Cho¹, Chin-Shiuh Shieh¹,
Hong-Jen Lee¹, and Fu-Min Fang^{2,*}

¹ National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan 807, ROC

² Chang Gung Memorial Hospital-Kaohsiung Medical Center, Chang Gung University
College of Medicine, Kaohsiung, Taiwan, ROC
fang2569@adm.cgmh.org.tw

Abstract. A successful adoption and adaptation of the particle swarm optimization (PSO) algorithm is presented in this paper. It improves the performance of Support Vector Machine (SVM) in the classification of incipient faults of power transformers. A PSO-based encoding technique is developed to improve the accuracy of classification. The proposed scheme is capable of removing misleading input features and, optimizing the kernel parameters at the same time. Experiments on real operational data had demonstrated the effectiveness and efficiency of the proposed approach. The power system industry can benefit from our system in both the accelerated operational speed and the improved accuracy in the classification of incipient faults.

Keywords: particle swarm optimization, incipient fault, classification.

1 Introduction

According to the IEC 599/IEEE C57 [1], degradations of the transformer oil and insulating materials produce gases, such as hydrogen (H_2), oxygen (O_2), nitrogen (N_2), methane (CH_4), ethylene (C_2H_4), acetylene (C_2H_2), ethane (C_2H_6), carbon dioxide (CO_2), and carbon monoxide (CO). They all can be detected from a DGA examination. In the presence of these gases, meaning corona or partial discharge, thermal heating and arcing may occur. The ratios between different gas concentrations, then, can be used to classify faults. However, any DGA approach is inherently imprecise and demands other supporting techniques to obtain reliable results. Some unidentified cases occurred in IEC 599, for example, when ratio codes calculated from actual DGA results did not correspond to any of the codes associated with a characteristic fault. One must then refer to Publication 60599, which contains an in-depth description of the five main types of faults usually found in electrical equipment in service. Publication 60599 classifies faults according to the main types that can be reliably identified by visual inspection of the equipment, after the fault has occurred in service [2]: (1) Partial discharges (PD) of the cold plasma (corona) type, (2) Low-energy

* Corresponding author.

discharges (D1), (3) High energy discharges (D2) with power follow-through, (4) Thermal faults below 300 °C if the paper has turned brownish (T1), and above 300 °C if the paper has carbonized (T2); (5) Thermal faults above 700 °C (T3).

The three basic gas ratios described in IEC 599 (C_2H_2/C_2H_4 , CH_4/H_2 , and C_2H_4/C_2H_6) are also used in IEC 60599 to identify the characteristic faults. The ratio limits have also been made more precise, in order to reduce the number of unidentified cases from around 30% in IEC 599 to nearly 0% in IEC 60599.

This paper proposes a method to improve the performance of Support Vector Machines (SVM) for fault diagnosis of power transformers by a particle swarm optimization. It improved the accuracy of classification by removing redundant input features which might cause confusion.

2 Particle Swarm Optimization

PSO algorithm optimizes an object function by conducting population-based search. The population consists of potential solutions, called particles, which are metaphor of birds in bird flocking. These particles are randomly initialized and then freely fly across the multi-dimensional search space. During the flying, every particle updates its velocity and position based on the best experience of its own and the entire population. The updating policy will drive the particle swarm to move toward region with higher object value, and eventually all particles will gather around the point with highest object value. The detail operation of PSO is as follows [3]:

Step 1. Initialization

The velocity and position of all particles are randomly set to within pre-specified or legal range.

Step 2. Velocity Updating

The velocities of all particles are updated according to the following rule:

$$\vec{v}_i \leftarrow w \cdot \vec{v}_i + c_1 \cdot R_1 \cdot (\vec{p}_{i,best} - \vec{p}_i) + c_2 \cdot R_2 \cdot (\vec{g}_{best} - \vec{p}_i) \quad (1)$$

where $\vec{p}_i$ and $\vec{v}_i$ are position and velocity of particle i , respectively; $\vec{p}_{i,best}$ and $\vec{g}_{best}$ are the position with best object value found so far by particle i and the entire population, respectively; w is a parameter controlling the dynamics of flying; R_1 and R_2 are random variables from the range [0,1]; c_1 and c_2 are factors used to control the related weighting of corresponding terms.

After the updating, $\vec{v}_i$ should be checked and clamped to pre-specified range to avoid violent random walking.

Step 3. Position Updating

Assuming unit time interval between successive iterations, the positions of all particles are updated according to the following rule:

$$\vec{p}_i \leftarrow \vec{p}_i + \vec{v}_i \quad (2)$$

After the updating, the $\vec{p}_i$ would also be checked and clamped to legal range to ensure legal solutions.

Step 4. Memory Updating

Update $\bar{p}_{i,best}$ and $\bar{g}_{best}$ when condition is meet.

$$\bar{p}_{i,best} \leftarrow \bar{p}_i \text{ if } f(\bar{p}_i) > f(\bar{p}_{i,best}), \bar{g}_{best} \leftarrow \bar{p}_i \text{ if } f(\bar{p}_i) > f(\bar{g}_{best}). \quad (3)$$

where $f(\bar{x})$ is the object function subject to maximization.

Step 5. Termination Checking

Repeats Step 2 to Step 4 until certain termination condition is met, such as predefined number of iterations is reached or failed to have progress for certain number of iterations. Once terminated, particle swarm optimization reports the $\bar{g}_{best}$ and $f(\bar{g}_{best})$ as its solution.

3 Review of Support Vector Machines

The idea of SVM was first proposed by Vapnik of Lucent Technology's Bell Laboratories [4]. In standard linear classification problem, suppose we have l given observations, each consisting of a pair of data: a data vector $x_i \in \mathbb{R}^n$, $i = 1, \dots, N$ and a class label $y_i \in \{+1, -1\}$ for each vector. The binary classification problem can be posed as follows:

$$\text{Minimizing} \quad F = \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^l \xi_i, \quad (4)$$

$$\text{subject to the constraints} \quad y_i ((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, l. \quad (5)$$

Where $y_i [\mathbf{w} \cdot \phi(x_i) + b] \geq 1$ comprises first the given constraints $\mathbf{w} \cdot \phi(x_i) + b \geq +1$ if $y_i = +1$, $\mathbf{w} \cdot \phi(x_i) + b \leq -1$ if $y_i = -1$, and C is used to weight the penalizing relaxation variables ξ_i and $(\cdot)$ denotes the inner product of $\mathbf{w}$ and $\mathbf{x}$ vectors.

In the case of SVMs used for binary classification, we use the decision function $sign(f)$, see (6). The following parameters need to be determined: the bias b , the number of support vectors m , the support vectors x_i and associated y_i values, $i = 1, 2, \dots, l$, as well as the Lagrangian multipliers α_i . The decision function has the following form:

$$y = sign(f(x)) = sign\left(\sum_{s.v.}^m \alpha_i^* y_i K(x_i, x_j) + b^*\right) \quad (6)$$

4 Proposed Method of Fault Diagnosis

In this paper, we demonstrate the efficiency of the PSO in the optimization design which was used for selection of the features and optimal classifier parameters (values of C , ξ and σ) of a SVM classifier with Radial Basis Function (RBF) kernel. The σ denotes the width of RBF kernel. The overall block diagram of PSO-based optimization algorithm is shown in Fig. 1.

For ANN learning, it was trained by using the back-propagation algorithm with adaptive learning strategy. Training was stopped through use of a validation set, with the stop occurring when the performance on the validation set started deteriorating rather than improving. For training of the SVM is carried out using the Kernel

Adatron (KA) algorithm. The advantage that the KA algorithm offers is that it allows the use of a validation set to stop training, and thus retains good generalization.

For comparison, training was carried out using first the ANN and then the same experiments were repeated with the SVM using both constant width RBF functions, where the σ parameter is found by iteration to one decimal place, and using average σ parameter, with a range of different scaling factors from 1 to 10. Three DGA features sets were used in the features extraction phase. We are concerned not only from the basic gas ratios of DGA data, but also their mean (μ), root mean square value (rms), variance (σ^2); and other higher order central moments [5]:

$$\gamma_n = \frac{E\{[y_i - \mu]^n\}}{\sigma^n}, n = 1, 10, \quad (7)$$

where E represents the expected value of the function. Total numbers of derived features are 36. We randomly extracted instances for learning sets, validation sets and test sets for corresponding phases. Traditionally k -fold cross-validation is used as an estimator of the generalization error [5]. In this study, a negative mean absolute percentage error (-MAPE) is used as the fitness function. The MAPE is as follows [6]:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{a_i - d_i}{a_i} \right| \times 100\% \quad (8)$$

where a_i and d_i represent the actual and diagnosis values, and N is the number of diagnosis periods. Based on fitness functions, particles with higher fitness values are more likely to yield a smaller MAPE value. Once procedures are terminated, particle swarm optimization reports the $\vec{g}_{best}$ and $f(\vec{g}_{best})$ as its solution.

At the end of learning stage, a set of features and parameters for a kernel function are obtained for the PSO-based SVM classifier. After learning phase, the classifier is ready to be used to classify new DGA pattern samples in classification phase.

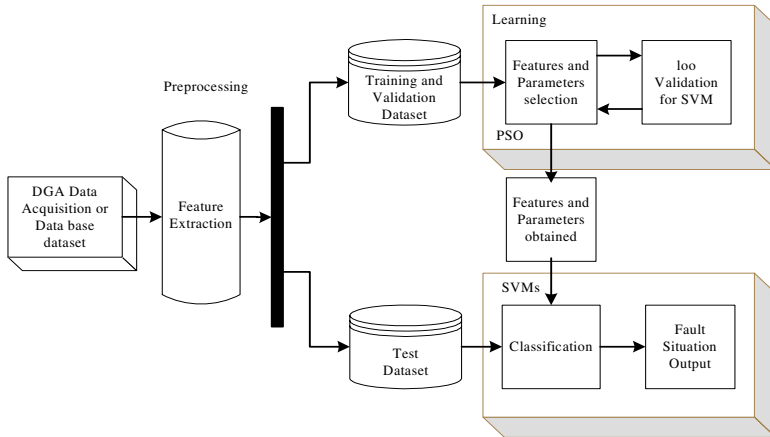


Fig. 1. Overall structure of proposed method

5 Experimental Results

Two sets of DGA data were compiled for the feasibility study of our proposed method. First data set consisting 134 gas records from IEC TC10 Database which have been discriminated into 6 classes based on IEC 60599 described values corresponding to different actual faults [7]. The second dataset contained 635 historical dissolved gas records from different transformers of the Taipower Company. These data are divided into three data sets: the training dataset, the validation dataset and the testing dataset.

5.1 Performance for the IEC TC 10 Dataset

Classification results are presented for stand alone ANN's and SVM's with/without features and parameter selection. For each case of stand alone ANN, the number of neurons in the hidden layer was set to 24; for stand alone SVM's, a constant width (σ) of 0.5 and tradeoff regularization parameter (C) of 100 was used. These values were chosen as the basis of initial trials of trainings.

Table 1 shows the classification results for stand alone ANN's and SVM's with the first IEC TC 10 dataset. In each case all features from the DGA signals are without CSA features and parameters optimization. The test success of stand alone ANN's was 83.6% and 90.4% for SVM's. In the Table 2 demonstrates the proposed method, where CSA optimization is applied. All 36 features were auto-selected from the corresponding input, the test success rate improved substantially for both ANN's and SVM's. The remaining features are 12 for ANN and 4 for SVM respectively. Test success was 95.3% for ANN's and 99.1% for SVM's. In each case, the training time of ANN was much higher than that of the SVM. The computational time (on a PC with Pentium IV processor of 2.4GHz and 512MB RAM) for training the classifiers are also shown in the tables.

Table 1. Performance without PSO optimization for the IEC TC 10 dataset

Classifier	Stand alone			
	No. of inputs	Training success (%)	Test success (%)	Training time (s)
ANN	36	78.7	83.6	16.385
SVM	36	81.8	90.4	4.937

Table 2. Performance with PSO optimization for the IEC TC 10 dataset

Classifier	With PSO			
	No. of inputs	Training success (%)	Test success (%)	Training time (s)
ANN	12	89.9	95.3	5.342
SVM	4	97.9	99.1	3.135

5.2 Performance for the Taipower Company Dataset

Table 3 shows the classification results for ANN's and SVM's with the Taipower Company data. The test success rate of stand alone ANN's were 82.9% and 89.7% for SVM's. With our proposed method, for ANN's, the number of neurons in the hidden layer was automatically selected from 10–36; and for SVM's, the RBF kernel width was automatically selected between 0.1–2.0, the test success rate improved substantially, 99.7% for ANN's and 100% for SVM's which were shown in Table 4. This is because of our method is capable of removing redundant input features that may be confusing the classifier, and hence to improve the generalization performance. The remaining features are 4 for both ANN and SVM classifiers respectively.

Table 3. Performance without PSO optimization for the Taipower Company dataset

Classifier	Stand alone			
	No. of inputs	Training success (%)	Test success (%)	Training time (s)
ANN	36	77.6	82.9	20.221
SVM	36	79.0	89.7	4.104

Table 4. Performance with PSO optimization for the Taipower Company dataset

Classifier	With PSO			
	No. of inputs	Training success (%)	Test success (%)	Training time (s)
ANN	4	100	99.7	6.445
SVM	4	100	100	2.676

6 Conclusions

In this paper, the selection of input features and the appropriate classifier parameters were optimized by using a PSO-based approach. Experiment results on real data demonstrate the effectiveness and high efficiency of the proposed approach. Without the incorporation of PSO, the classification accuracy of SVMs was better than of ANNs. With PSO-based selection, the performances of both classifiers were comparable at nearly 100%. We conclude that our method is not only able to find out optimal parameters but also minimize the number of features that SVM classifier should process and consequently maximize the classification rates of DGA.

References

1. IEEE Std.C57.104, *IEEE Guide for The Interpretation of Gases Generated In Oil-Immersed Transformers*, 1991.
2. M. Dong, "Fault diagnosis model power transformer based on statistical learning theory and DGA," in *Proceedings of the IEEE 2004 International Symposium on Electrical Insulation*, p85-88, 2004.

3. S.-C. Chu, C.-S. Shieh, and J. F. Roddick, "A tutorial on meta-heuristics for optimization," in J.-S. Pan, H.-C. Huang, and L.C. Jain (Eds.), *Intelligent Watermarking Techniques*, World Scientific Publishing Company, Singapore, Chapter 4, pp. 97-132, 2004.
4. V. Vapnik, S. Golowich, A. Smola, "Support vector method for function approximation, regression estimation, and signal processing" in M. Mozer, M. Jordan, T. Petsche (Eds), *Neural Information Processing Systems*, vol. 9, MIT Press, 1997.
5. L.B. Jack and A.K. Nandi, "Fault detection using SVM and artificial neural networks: augmented by genetic algorithms," *Mechanical Systems and Signal Processing*, pp. 373-390, 2002.
6. P.F. Pai and W.C. Hong, "Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms", *Electric Power System Research*, vol. 74, pp. 417-425, 2005.
7. M. Duval, "Interpretation of Gas-In-Oil analysis using new IEC Publication 60599 and IEC TC 10 Databases," *IEEE Electrical Insulation Magazine*, vol. 17, no. 2, 2001.

AntTrend: Stigmergetic Discovery of Spatial Trends

Ashkan Zarnani¹, Masoud Rahgozar², Caro Lucas³, and Azizollah Memariani⁴

^{1,2,3} Database Research Group (CIPCE), Electrical and Computer Engineering Department,
University of Tehran, no. 9 Alley, Amir Abad St., Tehran, Iran

a.zarnani@ece.ut.ac.ir, rahgozar@ut.ac.ir, lucas@ipm.ir

⁴ Department of Industrial Engineering, Bu-Ali Sina University, Hamedan, Iran
a.memariani@basu.ac.ir

Abstract. Large amounts of spatially referenced data have been aggregated in various application domains such as Geographic Information Systems (GIS), banking and retailing that motivate the highly demanding field of spatial data mining. So far many beneficial optimization solutions have been introduced inspired by the foraging behavior of ant colonies. In this paper a novel algorithm named AntTrend¹ is proposed for efficient discovery of *spatial trends*. AntTrend applies the emergent intelligent behavior of ant colonies to handle the huge search space encountered in the discovery of this valuable knowledge. Ant agents in AntTrend share their individual experience of trend detection by exploiting the phenomenon of *stigmergy*. Many experiments were run on a real banking spatial database to investigate the properties of the algorithm. The results show that AntTrend has much higher efficiency both in performance of the discovery process and in the quality of patterns discovered compared to non-intelligent methods.

Keywords: Spatial Data Mining, Ant Colony Optimization, Spatial Trends.

1 Introduction

Many organizations have collected large amounts of spatially referenced data in various application areas such as geographic information systems (GIS), banking and retailing. These are valuable mines of knowledge vital for strategic decision making and motivate the highly demanding field of spatial data mining; i.e., the discovery of interesting, implicit knowledge from large amounts of spatial data [11].

So far many data mining algorithms have been developed to be applied on spatial databases. In [11, 17] spatial association rules are defined and algorithms are proposed to efficiently exploit the concept hierarchy of spatial predicates in the mining process. In [8, 12] algorithms are designed for the classification of spatial data. Shekhar *et al.* further improved spatial classification in [15] and also introduced algorithms for the discovery of co-location patterns [10]. Also spatial clustering was studied in [13]. Spatial trends are one of the most valuable and comprehensive patterns potentially found in a spatial database. In spatial trend analysis, patterns of change of some

¹ This work was supported in part by the TAKFA Grant Program and Mellat Bank R&D Center.

non-spatial attributes in the neighborhood of an object are explored [6, 7]; e.g. moving towards north-east from the city center, the average income of the population increases (confidence 82%).

A path in a spatial graph is considered to be a valid trend if the confidence of regression on non-spatial values of its vertices based on their distance from a start object is above a user given threshold [6, 7]. This is called local trend in [6, 7]. Ester *et al.* studied the task of spatial trend discovery by proposing an algorithm which applies a general clustering method [9]. This algorithm was further improved in [7] by exploiting the database primitives for spatial data mining introduced in [6, 8]. In this later algorithm, first a specified start object o is given by the user. Then it examines every possible path in the neighborhood graph beginning from o to check its regression confidence. But in this approach the search space soon becomes tremendously huge by increasing the size of neighborhood graph and makes it impossible to do a full search. In order to prune the search space it applies a heuristic which will be shown here that it is restrictive.

Many solutions for NP-Complete search and optimization problems have been developed inspired by the cooperative foraging behavior of ant colonies [2]. However, less attention has been given to this powerful inspiration from the nature in the tasks of spatial data mining. In this research, we introduce a new spatial trend detection algorithm named AntTrend. AntTrend uses the phenomenon of *stigmergy*; i.e. indirect communication of simple agents via their surrounding environment, observed in real ant colonies [1]. It also combines this behavior with a new guiding heuristic that is shown to be effective. We succeeded to handle the non-polynomial growth of the search space, and at the same time retain the discovery power of the algorithm. This was achieved by letting each ant agent cooperatively exploits the colony's valuable experience. Also in contrast with the algorithm proposed in [7], AntTrend is not dependent on the user's initial inputs. It does not get a specified start object from the user nor needs to input a pruning threshold. This brings ease of use and wider applicability to AntTrend as its efficiency and performance is independent of the user. We have conducted many experiments on a real banking spatial database to analyze the properties of AntTrend and to compare it with the algorithm proposed in [7], which is widely accepted and used.

2 Spatial Trend Detection

Some spatial relations between objects (called neighborhood relations) are formally defined to be used in spatial data mining. These include direction, metric and topological relations (e.g. object A is south-east of object B) [6]. Based on these relations the notions of neighborhood graph and neighborhood path are defined as follows [8]:

Definition 1: let neighbor be a neighborhood relation and DB be a database of spatial objects. Neighborhood graph $G = (N, E)$ is a graph with nodes $N = DB$ and edges $E \subseteq N \times N$ where an edge $e = (n_1, n_2)$ exists iff $neighbor(n_1, n_2)$ holds.

Definition 2: A *neighborhood path* of length k is defined as a sequence of nodes $[n_1, n_2, \dots, n_k]$, where $neighbor(n_i, n_{i+1})$ holds for all $n_i \in N$, $1 \leq i < k$.

As we have the location dimension in a spatial database, one useful potential pattern could be the change of a non-spatial attribute with respect to its distance from a reference object. E.g. beginning from a trading center in the city and moving on a specific highway towards the west, the unemployment rate grows. Having available the desired neighborhood graph, a spatial trend is defined in [7] as follows:

Definition 3: A spatial trend is a path on the neighborhood graph with a length k of nodes such that the confidence of regression on its nodes data values and their distance from the start node (see Figure 1.a) is above a threshold given by the user.

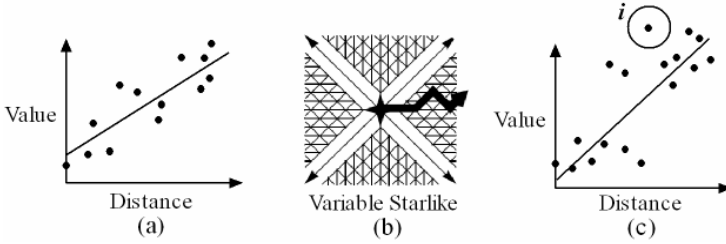


Fig. 1. (a) Regression line for a trend (b) A direction filter (c) Missing a trend in node i

Our example spatial database contains the agency locations of a national bank and their various (non-spatial) financial data like the count and balance of different kinds of accounts. A map of these points and a sample trend are provided in Figure 2.

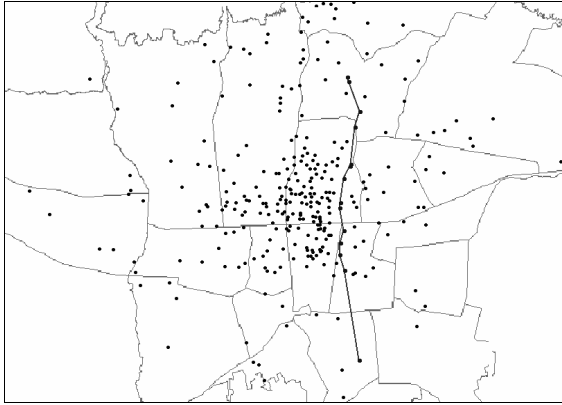


Fig. 2. The City Regions with Bank Points

As an example we may need to find trends of "the number of long term accounts" in the bank agencies starting from any arbitrary agency points. Having discovered such trends, we can try to explain them by some spatial attributes. As an example a trend could be matching a road or a highway. We can also check if there are any matching trends on the same path but in other thematic layers such as demographic or land use layers. A trend can predict the data value of a new point on its path with a

reliability fraction equal to its regression confidence. A desired informative spatial trend pattern would not be crossing the space in an arbitrary manner. So a direction filter is applied when forming the path of a trend being examined. Ester *et al.* used some direction filters in [7] like the variable starlike filter depicted in Figure 1.b. Here we have applied a direction filter of our own (see subsection 4.4).

To discover the trends in a neighborhood graph by the algorithm proposed in [7], having ' a ' feasible nodes on average to extend a path, we would have to meet a^n paths to examine their regression confidence, where n is the maximum trend length. It's impossible to examine this amount of paths even for not much large values of n (e.g. $n=20$). This condition gets worse when the user has not any specific start object in mind, and expects the algorithm to check different start objects. So for higher efficiency, the algorithm allows a path to be extended further by the next set of feasible nodes only if its current confidence is not below the given threshold. This is continued until we reach the maximum length. This heuristic rule forces the search space to become smaller but it can easily miss a high confidence trend if its confidence is below the threshold somewhere in the middle of its path. In Figure 1.c a sample of this situation is shown. The algorithm will stop path extension when it is in node i because the regression confidence of the path from the first node to i is below the threshold. However this path would have a confidence much higher than the threshold if it is continued (i.e. not stopped).

In AntTrend, different ant agents will start searching from different start points so that there is no need to ask the user for a specific start object. Also as the experience of each individual ant is shared in the colony by means of *stigmergy*, the algorithm will have much higher performance in the discovery of spatial trends. As a result of this improvement we can remove the restricting assumption explained above.

3 Ant Colonies for Search and Optimization

Ant Colony Optimization (ACO) is a widely-used meta-heuristic inspired by the behavior of real ant colonies in the nature. Ant colonies can intelligently solve complex discrete problems (e.g. finding shortest path) although their individuals are so simple and never intelligent enough to solve such problems on their own [3, 5]. This approach was first applied by Dorigo *et al.* on the traveling salesman problem in the algorithm called Ant System [4]. The interested reader is referred to [2, 4, 5] for the basic concepts and applications of ACO.

ACO has been recently used in many areas [2, 5] including data mining [14]. However considering the challenges faced in the problem of spatial trend detection (see section 2), we can notice that ACO provides efficient properties in spatial trend discovery too. Firstly, as the definition of the problem suggests, ant agents can search for a trend starting from their own start point in a completely distributed manner. This omits the need to ask the user for a start object. Secondly, the pheromone trails can help the ants to exploit the trend detection experience of the whole colony. This guides the search process to converge to a better promising subspace that contains more potential high quality trend patterns. Finally, some measures of attractiveness can be defined for each of the feasible spatial nodes to extend a path. This can efficiently guide the trend discovery process of each ant.

4 AntTrend Algorithm

As mentioned before, we have developed a trend detection algorithm with the ACO approach named AntTrend. In this algorithm different ant agents start from different points of the graph to search for increasing spatial trends (i.e. the slope of the regression line is to be positive). Note that this assumption is not restricting because each decreasing trend is also an increasing one when started from the last node. In this section we explain AntTrend algorithm within the framework of ACO algorithms.

4.1 Graph Construction

In order to define a neighborhood graph in our spatial database we assume that the bank points P_i and P_j are connected with two directed edges E_{ij} and E_{ji} iff: $\text{Distance}(i, j) < \text{Max-Distance}$ and there is no other point P_k in the Hallow-Area of P_i and P_j namely H_{ij} .

H_{ij} is the area where for any point P_k in this area:

- The distance between P_k and the spatial line connecting P_i and P_j is less than a given maximum value, namely *Hallow-Distance*.
- The angle between the lines of (P_k, P_i) and (P_k, P_j) with respect to the line of (P_i, P_j) is less than a given maximum value namely *Hallow-Angle*.

This means that if there is to be a spatial trend where P_j comes after P_i , P_k must be also present in that, making a spatial sequence of P_i , P_k and P_j . Figure 3 graphically shows how the *Hallow-Area* is formed.

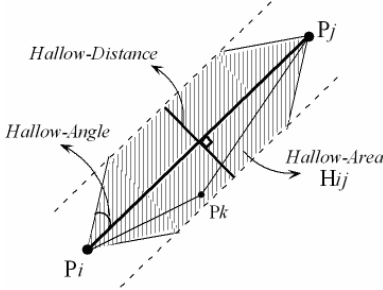


Fig. 3. Hallow-Area Boundaries

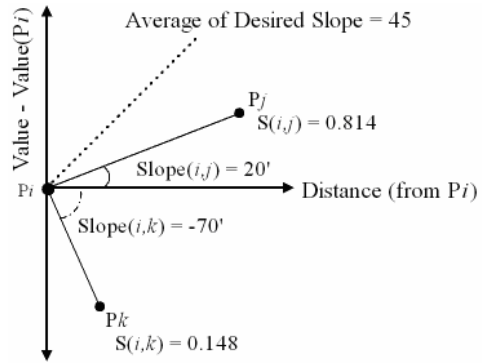


Fig. 4. Slope Heuristic

The parameters of the *Hallow-Area* are tuned based on the experts' knowledge in the application. Finally a direction is assigned for each spatial edge with respect to its angle with the horizontal line connecting west to east (eight possible directions). This direction is used for filtering in the process of building a candidate trend.

4.2 Pheromone Trail Definition

The quality of an ACO application depends very much on the definition assigned to the pheromone trail. It is crucial to choose a definition conform to the nature of the problem. Spatial trend discovery is an ordering problem in which the aim is to put the different spatial points in a certain order. As previously mentioned, the ants will search for increasing trends only, adding the direction of an edge to the properties of the laid pheromone. A pheromone trail value $\tau_{i,j}$ is considered for each directed edge $E_{i,j}$ of the neighborhood graph making the pheromone matrix asymmetric. When an ant is in node P_i and selects P_j as the next node, its pheromone is laid on the $E_{i,j}$ (and not E_{ji}). Thus $\tau_{i,j}$ encodes the favorability of selecting P_j when in P_i to form a high quality increasing spatial trend.

4.3 Heuristics for Spatial Trend Discovery

Another important feature of an ACO implementation is the choice of a good heuristic to incrementally build possible solutions. This heuristic will be used in combination with the pheromone information.

We applied two heuristics to guide the discovery of spatial trends. Firstly, closer nodes are preferable to the ones far from the current node, as they are more likely to be correlated. Secondly, we would like the value of the next node to match better with an increasing regression model (here linear regression). To apply this second heuristic we consider the slope of the line where its x coordinate is the distance between the two nodes and its y coordinate is their value targeted in trend detection (see Figure 4).

The more this line's slope is near to 45 degrees the more preferable it is. In this way there will not be much difference between the heuristic values of nodes having a positive slope. However, negative slope nodes will have a smaller desirability for selection, which decreases by increasing the negative slope. The heuristic value for selecting node P_j from node P_i will be calculated by the following formulas:

$$D_{i,j} = 1 / \text{Distance}(i, j) \quad (1)$$

$$S_{i,j} = 1 - \left| \frac{\text{Slope}(i, j) - 45}{135} \right| \quad (2)$$

$$\eta_{i,j} = S_{i,j}^{\beta} \cdot D_{i,j}^{\gamma} \quad (3)$$

Where β and γ are the relative importance of slope heuristic and distance heuristic, respectively. $\eta_{i,j}$ is the attractiveness of P_j for selection when in P_i . In Figure 4 two sample nodes (P_j and P_k) and their slope heuristic values with respect to P_i are shown.

4.4 Building Candidate Trends

The pheromone trail and the heuristic defined above will now be used by the ants to search the graph. Each ant agent starts from a node and forms a candidate trend by iteratively selecting a new node. Each ant k has a memory M^k to store the information of its current path such as its nodes and d_k which is the direction of the trend, used in direction filtering. The value of d_k is set to the direction of the second node with respect to the first node that ant k has chosen. Ant k stochastically selects the next node by assigning a probability $P^k_{i,j}$ to select P_j when in P_i by the following formula:

$$P_{i,j}^k = \begin{cases} \frac{[\tau_{i,j}]^\alpha \cdot [\eta_{i,j}]}{\sum_{l \in allowed_k} [\tau_{i,l}]^\alpha \cdot [\eta_{i,l}]} & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Where α is the relative importance of pheromone trail and $allowed_k$ is the set of nodes that are adjacent to P_i and have passed the direction filter. The filter we have used here accepts new directions to be the same as d_k or rotated one step either clockwise or counterclockwise. As an example if d_k is South-East then the nodes with directions South-East, South or East with respect to P_i will be in $allowed_k$.

The addition of a node is done by all of the ants independently and in parallel. This process is repeated until the number of nodes in the candidate trends reach to *TrendLength* which is an integer input of the algorithm, given by the user.

4.5 Updating the Pheromone Trail

To update the pheromone trail information, we consider the quality of the trends. This quality is evaluated by the coefficient of determination (r^2) in the linear regression. This value is a fraction between 0.0 and 1.0, and has no units. The better you can predict Y from X , the nearer is this value to one. In an iteration of AntTrend (called a cycle) m ants form their candidate trends with *TrendLength* nodes. At the end of a cycle the pheromone matrix is updated by the following formulas:

$$\tau_{i,j}(t+1) = \rho \cdot \tau_{i,j}(t) + \Delta \tau_{i,j} \quad (5)$$

$$\Delta \tau_{i,j} = \sum_{k=1}^m \Delta \tau_{i,j}^k \quad (6)$$

$$\Delta \tau_{i,j}^k = \begin{cases} Q \times confidence(k) & \text{if ant } k \text{ uses edge}(i,j) \text{ in its trend} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Where $(1 - \rho)$ is the factor of trail evaporation between two cycles. This lets the colony to avoid unlimited accumulation of trials over some component. *Confidence(k)* is the r^2 value of the regression for the candidate trend detected by ant k and Q is a constant.

5 Experimental Results

We conducted some experiments on our spatial database to compare AntTrend with the algorithm proposed in [7] and also to analyze the execution of our proposed algorithm. In Table 1 the values of the graph construction parameters and properties of the respective constructed neighborhood graph are given. There were 300 agencies and the final graph had 1535 edges between them. The neighborhood relation parameters were chosen based on our banking application in hand. Also for the non-spatial values of the nodes we aimed to discover their spatial trends (i.e. the number of long term accounts in each bank agency) some statistics are given in table 2.

Table 1. Neighborhood Graph parameters and properties

Construction Parameters					Obtained Graph Properties			
Nodes	Max-Distance	Hallow-Angle	Hallow-Distance	Candidate Edges	Edges	Min. Degree	Max. Degree	Avg. Degree
300	6000m	80°	300m	43336	1535	2	30	10.23

Table 2. Node values

Min.	Max.	Avg.	Std. Deviation
1	39025	4623	15631

Table 3. Two different setups for AntTrend

	α	β	$1-\rho$
Setup1	1	5	0.1
Setup2	5	1	0.6

To discover the trends of length 10 by AntTrend, we gained the best results by putting two ants in each node and the values of α , β , γ and ρ being respectively equal to 1, 4, 0.3 and 0.5 (we have used these values for all of the experiments if not mentioned). We have considered a path of nodes with its confidence (r^2) over 75% as a trend. As a comparison between AntTrend and the algorithm developed by Ester *et al.* [7], Figure 5 shows the number of trends that the two algorithms find when a certain equal number of paths in the graph have been examined in an equal time.

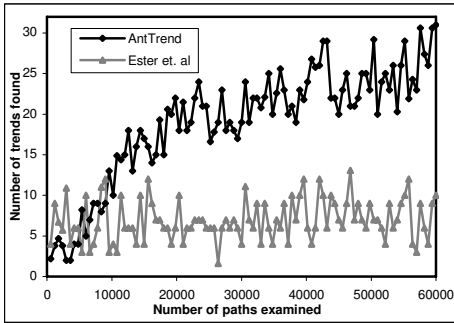


Fig. 5. Comparison between AntTrend and Ester's Algorithm

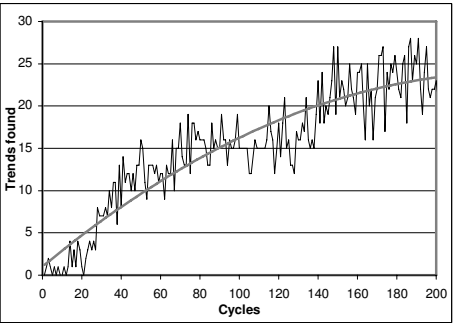


Fig. 6. Improvement in Trend Discovery with *Trend Length=15*

As can be seen the algorithm proposed by Ester does not use any dynamic guiding heuristic and its performance will not change as the number of examined paths increase. But AntTrend will improve its trend discovery as the colony aggregates its population's experience gained from the previous cycles and consequently outperforms the other algorithm drastically. Figure 6 shows how the algorithm improves its discovery power in search for trends of length 15. The regression line for the graph in this Figure confirms a smooth increase in the number of trends discovered in each cycle. There is also an evolution observed in the average confidence of paths examined in every cycle (see Figure 7) where in each cycle 600 paths are checked. The higher average confidence even in the beginning cycles approves the effectiveness of the guiding heuristics used. The trade off between importance of

trail intensity and heuristic value is a crucial one in an application of ant colony optimization and can highly affect its performance [2, 4]. To analyze this trade off in AntTrend, the algorithm was run in two different setups given in table 3 to discover trends of length 10 (i.e. *TrendLength*=10).

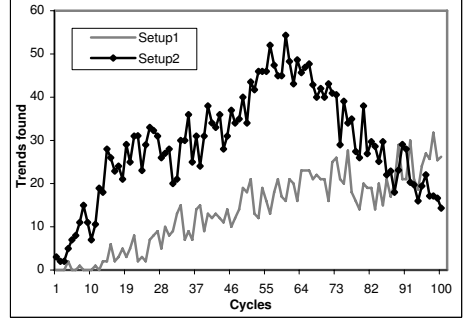
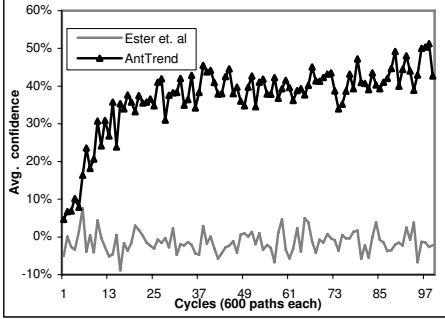


Fig. 7. Evolution of Average Trend Confidence **Fig. 8.** Comparison between two different parameter setups

Considering Figure 8, in setup1 the importance of trail intensity is less than heuristic and as the trail is updated in consequent cycles it guides the ants to better paths. But in setup2 the importance of trail intensity is much higher. Also the evaporation factor is higher so the pheromone trails on edges that are not reinforced decrease faster. Hence, the search concentrates earlier around the trends discovered so far and improves faster. However, after a number of cycles the ants show a *stagnation* behavior [4], finding less new trends in next cycles due to high pheromone values in some few edges.

6 Conclusions

In this paper, we proposed a new application of ant colony optimization for the task of spatial trend discovery introducing a novel algorithm named AntTrend. This algorithm applies the emergent intelligent behavior of ant colonies to handle the huge search space encountered in the discovery of this invaluable knowledge. We proposed two heuristics for node selection by the ants shown to be effective. AntTrend is also independent of the user as it does not need to input a start node. It also does not apply a user given threshold in the discovery process and consequently will not miss some valid trends. The experiments run on a real banking spatial database show that AntTrend outperforms the current approaches in performance and pattern quality. The high scalability of the algorithm makes it applicable on large spatial databases. In our future research we will further improve the heuristics and investigate the use of some modified versions of ACO like MMAS [16] in spatial trend detection. Currently, a limitation of AntTrend is that it searches for trends of a certain length. We plan to tackle this problem by integrating the search process for trends with different lengths.

References

1. Dorigo M., Bonabeau E., Theraulaz G.: Ant Algorithms and Stigmergy. *Future Generation Computer Systems*. vol. 17, no.8 (2000) 851–871.
2. Dorigo M., Di Caro G.: Ant Algorithms for Discrete Optimization. *Artificial life*. vol. 5 no. 2 (1999) 137–172.
3. Dorigo M., Di Caro G.: Ant Colony Optimization: A New Meta-Heuristic, *Proc. Congress on Evolutionary Computation*. (1999) 1470–1477.
4. Dorigo M., Maniezzo V., Colorni A.: The Ant System: Optimization by a Colony of Co-operating Agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 26, no.1 (1996) 29–41.
5. Dorigo M., Stützle T.: The Ant Colony Optimization Meta-Heuristic: Algorithms, Applications and Advances. In F. Glover and G. Kochenberger, *Handbook of Metaheuristics* Kluwer Academic Publishers (2002)
6. Ester M., Frommelt A., Kriegel H.P., Sander J.: Spatial Data Mining: Database Primitives, Algorithms and Efficient DBMS Support. *International Journal of Data Mining and Knowledge Discovery*, Kluwer Academic Publishers, vol. 4, no.2/3 (2000) 193–217.
7. Ester M., Frommelt A., Kriegel H.P., Sander J.: Algorithms for Characterization and Trend Detection in Spatial Databases. *Proc. 4th International Conf. on Knowledge Discovery and Data Mining*, (1998) 44–50.
8. Ester M., Kriegel H.P., Sander J.: Spatial Data Mining: A Database Approach. *Proc. 5th International Symp. On Large Spatial Databases*. (1997) 320–328.
9. Ester M., Kriegel H.P., Sander J., Xu X.: Density-Connected Sets and Their Application for Trend Detection in Spatial Databases. *Proc. 3rd International Conf. on Knowledge Discovery and Data Mining*. (1997) 44–50.
10. Huang Y., Shekhar S., Xiong H.: Discovering Spatial Co-location Patterns from Spatial Datasets: A General Approach. *IEEE Transactions on Knowledge and Data Eng.* vol.17, no.12 (2004) 1472–1485.
11. Koperski K., Han J.: Discovery of Spatial Association Rules in Geographic Information Databases. *Proc. 4th Int. Symp. on Large Spatial Databases* (1995) 47–66.
12. Koperski K., Han J., Stefanovic N.: An Efficient Two-step Method for Classification of Spatial Data. *Proc. International Symp. On Spatial Data Handling* (1998) 320–328.
13. Ng R., Han J.: CLARANS: A Method for Clustering Objects for Spatial Data Mining. *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5 (2005) 1003–1017
14. Parpinelli R.S., Lopes H.S., Freitas A.A.: Data Mining with an Ant Colony Optimization Algorithm. *IEEE Transactions on Evolutionary Computation*, vol. 6, no.4 (2002) 321–332.
15. Shekhar S., Schrater P., Vatsavai W. R., Wu W., S. Chawla: Spatial Contextual Classification and Prediction Models for Mining Geospatial Data. *IEEE Transactions on Multimedia*, vol. 2, no.4 (2002) 174–188.
16. Stützle T., Hoos H. H.: MAX-MIN Ant System. *Future Generation Computer Systems*, vol. 17, no. 9 (2000) 851–871.
17. Wang L., Xie K., Chen T., Ma X.: Efficient Discovery of Multilevel Spatial Association Rules Using Partitions. *Information and Software Technology*, Vol. 47, no. 13 (2005) 829–840.

Genetic Algorithm Based Approach for Multi-UAV Cooperative Reconnaissance Mission Planning Problem

Jing Tian¹, Lincheng Shen¹, and Yanxing Zheng²

¹ College of Mechatronic Engineering and Automation,
National University of Defense Technology, Changsha, China
jingtian@nudt.edu.cn

² Beijing Institute of System Engineering, Beijing, China

Abstract. Multiple UAV cooperative reconnaissance is one of the most important aspects of UAV operations. This paper presents a genetic algorithm(GA) based approach for multiple UAVs cooperative reconnaissance mission planning problem. The objective is to conduct reconnaissance on a set of targets within predefined time windows at minimum cost, while satisfying the reconnaissance resolution demands of the targets, and without violating the maximum travel time for each UAV. A mathematical formulation is presented for the problem, taking the targets reconnaissance resolution demands and time windows constraints into account, which are always ignored in previous approaches. Then a GA based approach is put forward to resolve the problem. Our GA implementation uses integer string as the chromosome representation, and incorporates novel evolutionary operators, including a subsequence crossover operator and a forward insertion mutation operator. Finally the simulation results show the efficiency of our algorithm.

1 Introduction

Using a fleet of sensor-bearing Unmanned Aerial vehicles (UAVs) to conduct air reconnaissance is one of the excellent applications of UAVs, and remains key activities in military operations. Multi-UAV cooperative reconnaissance mission planning problem(CRMPP) can be described as that, given a set of different targets, mission plans should be quickly made for UAVs that meet the needs of the commanders and can deal with the real-world constraints such as time windows, imagery requirements(reconnaissance resolution), and UAVs with different capabilities(i.e. imagery equipment, speed and range).

CRMPP comprises two parts, the routing and the scheduling of UAVs. The routing, like the Travelling Salesman Problem(TSP), is concerned with finding an optimal reconnaissance sequence of the targets. While the scheduling, like the Job Scheduling Problem(JSP), specifies the reconnaissance time on the targets. Therefore, CRMPP should address not only the spatial but also the temporal aspects of UAVs movement. Some literatures have been proposed for the

cooperative reconnaissance problems using multiple UAVs. Ryan[1] formulated the problem as a multiple travelling salesman problem(TSP) with the objective of maximizing expected target coverage and solved by tabu search algorithm. Hutchison[2] proposed a method beginning with an arbitrary number of targets randomly distributed over a given geographic area and the base for the UAVs. The operations area is divided into identical sectors and targets in one sectors are allocated to a given UAV. Then classical TSP defined to find the shortest path connecting each target for an UAV is solved using simulated annealing algorithm. Ousingsawat[3] divided the problem to two issues, minimum-time trajectory problem and task assignment problem. A* search is used to find the shortest path between two points, and the possible path combinations are listed for all UAVs, then the task assignment problem is expressed as a mixed integer linear program(MILP) to solve.

While in the previous studies, two important factors of cooperative reconnaissance problem are often ignored. One is the reconnaissance resolution. When UAV conducts reconnaissance on a target, the imaging resolution of the sensors equipped in the UAV will affect the reconnaissance results. For a certain ground target, the imaging resolution on it should reach a certain level for the commander to recognize it or classify it. If the resolution didn't reach the required level, the reconnaissance is useless. Another factor is the time windows for UAVs conducting reconnaissance on the targets. The information about the target should be timely and reflect the situation in a certain period. If the UAVs are tardy, the reconnaissance is also failure. Therefore, when making mission plans for the UAVs, we should consider the resolution demands and time windows of the targets to ensure that the reconnaissance is effective.

This paper presents a mathematical model for cooperative reconnaissance mission planning problem(CRMPP). The model takes the reconnaissance resolution demands and time windows of the targets into account, which are absent in previous studies. Then a novel genetic algorithm based approach is presented for CRMPP. Integer string representation of the chromosome is introduced, and new crossover and mutation operators are designed according to the specifics of CRMPP.

The rest of the paper is organized as follows. Section 2 presents the mathematical model of CRMPP. The GA based approach for CRMPP is presented in section 3, including the chromosome representation, initial population creation, evolutionary operators etc.. Section 4 shows the simulation results and 5 concludes the paper.

2 Problem Formulation

While in the previous formulation of the problem, such as TSP in [1][2] and MILP in [3], reconnaissance resolution and time windows of the targets, which are important factors, are often ignored. We present a mathematical formulation of the problem taking the two factors into account.

The reconnaissance targets set is denoted by $T_0 = \{1, 2, \dots, N_T\}$, and $T = \{0, 1, \dots, N_T\}$ is the extended targets set, where N_T is the number of targets and 0 indicates the base that UAVs depart from and return to. R_i is the reconnaissance resolution demand of target $i \in T_0$. Each target $i \in T_0$ has a time window $[e_i, l_i]$, where e_i is the earliest time allowed for reconnaissance on it and l_i is the latest time. T_i is the time to begin reconnaissance on target i . An UAV may arrive before e_i , which incurs the waiting time w_i until reconnaissance is possible. No UAV may arrive past l_i . $V = \{1, 2, \dots, N_v\}$ denotes the UAVs set, where N_v is the number of UAVs. The reconnaissance resolution is r_v for UAV v , and TL_v is the maximum travel time permitted. s_{vi} denotes the time duration for UAV v conducting reconnaissance on target i . The routes set between target pairs is denoted by $A = \{(i, j) | i, j \in T, i \neq j\}$, which can be calculated by classical routing algorithms such as A*. Each route $(i, j) \in A$ is associated with a distance Dis_{ij} denoting the length of (i, j) and a travel time t_{vij} denoting the travel time for UAV v between target i and j . The decision variable of CRMPP is

$$x_{vij} = \begin{cases} 1 & \text{if route } (i, j) \text{ is used by UAV } v \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The CRMPP is stated as follows:

$$(\text{CRMPP}) \quad \min \sum_{vij} f_{vij} x_{vij} \quad (2)$$

Where $f_{vij} = \alpha Dis_{ij} + \beta(T_i + s_{vi} + t_{vij} + w_j)$

Subject to:

$$\sum_{j \in T_0} x_{v0j} = 1, \sum_{j \in T_0} x_{vj0} = 1, \forall v \in V \quad (3)$$

$$\sum_{v \in V} \sum_{i \in T} x_{vij} = 1 \quad \forall j \in T_0 \quad (4)$$

$$\sum_{v \in V} \sum_{j \in T} x_{vij} = 1 \quad \forall i \in T_0 \quad (5)$$

$$e_i \leq T_i \leq l_i \quad \forall i \in T_0 \quad (6)$$

$$\text{If } x_{vij} = 1, \text{ then } T_i + s_{vi} + t_{vij} + w_j = T_j \quad (7)$$

$$w_j = \max(0, e_j - (T_i + s_{vi} + t_{vij})) \quad (8)$$

$$\sum_{(i,j) \in A} x_{vij} t_{ij} + \sum_{(i,j) \in A} x_{vij} s_{vj} + \sum_{(i,j) \in A} x_{vij} w_j \leq TL_v \quad \forall v \in V \quad (9)$$

$$\text{If } x_{vij} = 1, \text{ then } r_v \leq R_j \quad (10)$$

The objective is to minimize the total UAV travel distance and time subject to many constraints, including the travel time, arrival time feasibility, and reconnaissance resolution demands. Equation (3) specifies that each UAV should

begin at the base and return to the base at the end of the mission. Equation (4) and (5) ensure that each target can be reconnoitred by one and only one UAV. The time feasibility constraints are defined in equations (6)-(9). Inequality (6) imposes the time window constraints. Waiting time is the amount of time that an UAV has to wait if it arrives at a target location before the earliest time for that target. Inequality (9) states that a feasible solution for the CRMPP conducts reconnaissance on all the targets without exceeding the maximum travel time of UAVs. Inequality (10) restricts that the UAV conducting reconnaissance on a target should satisfy the reconnaissance resolution demands of the target.

3 GA Based Approach for CRMPP

CRMPP is something like vehicle routing problem with time windows(VRPTW). VRPTW has been proved to be NP-hard problems[4]. Due to the inherent complexity of the problem, search methods based upon heuristics are most promising for solving the problem. In particular, Genetic Algorithm(GA), a stochastic search based on the mechanics of natural selection and genetics, attracts much attention for solving VRPTW [5][6][7]. In this section, we study the GA based approach for resolving CRMPP.

3.1 Chromosome Representation

The first step to apply GA to a particular problem is selecting an internal string(chromosome) representation for the solution space. The choice of this component is one of the critical aspects to the success/failure of GA. Considering special characters of CRMPP, we make the following assumptions (the assumptions will not simplify the problem):

(1) All the UAVs are sorted ascending according to the reconnaissance resolutions, i.e. for $V = \{1, 2, \dots, N_V\}$, $r_1 < r_2 < \dots < r_{N_V}$ holds. (The smaller the r_v , the higher the resolution.)

(2) All targets in T_0 are clustered according to their reconnaissance resolution demands. For target $t \in T_0$, if $r_j < R_t < r_{j+1}$, then $t \in C_j$, $j = 1, 2, \dots, N_V$, which means that UAV $1, 2, \dots, j$ can satisfy the reconnaissance resolution demands of target t , while UAV $j + 1, \dots, N_V$ can not.

In our approach, a chromosome is given by an integer string of length $(N_T + N_V + 1)$. As shown in Fig.1, a chromosome consists of N_V subsequences, and each subsequence represents a reconnaissance target sequence for an UAV. In the following parts, the chromosome is represented as $S = \{sr_1, sr_2, \dots, sr_{N_V}\}$, where sr_i is the reconnaissance target sequence for UAV i . According to the above assumptions, all targets in C_j could appear in one of the subsequences in $\{sr_i | i \leq j\}$ and could not be in any of the subsequences in $\{sr_i | i > j\}$. Thus the solution that the chromosome encodes can satisfy the constraints defined by equation(10). Therefore, the chromosome is a strict ordered structure, in which the targets in the same subsequence can not exchange positions and different subsequences can not exchange positions either.

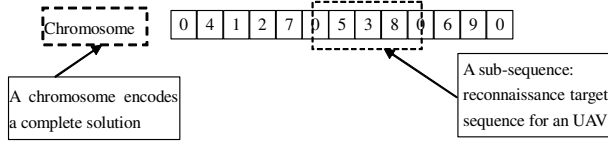


Fig. 1. Data structure of chromosome representation

3.2 Creation of Initial Population

The creation of initial population is an important component in genetic algorithm. We generate a mix population comprises feasible solutions and random solutions. First, we construct a feasible solution S_0 , and its neighbors $S \in N(S_0)$ as a portion of the initial population. The rest solutions of the population are generated randomly. The reason for constructing this mix population is that, a population of members entirely from the same neighborhood gives up the opportunity of exploring the whole regions; and a completely random population may be largely infeasible hence taking a long time to coverage.

First, an algorithm *Initial-Feasible-Solution-Generation*(IFSG) is proposed to construct initial feasible solution for the problem. The detail steps of IFSG are described as Fig.2.

Algorithm 1 *Initial-Feasible-Solution-Generation*)

- (1) Set $S_0 = sr_1, \dots, sr_{N_v}$, $sr_i = \text{NULL}$, $i = 1, \dots, N_v$
Targets Set: $C_1, C_2, \dots, C_{N_v}$
- (2) $k = 1$
- (3) While C_k is not empty
For all targets t in C_k
Insert t to subsequence set $SS_t = \{sr_1, \dots, sr_k\}$
Delete t from C_k
- (4) $k = k + 1$
- (5) If $k \leq N_v$, go to step(3)
Else output $S_0 = sr_1, \dots, sr_{N_v}$. Exit.

Fig. 2. Initial-Feasible-Solution-Generation

In the IFSG, the steps that insert a target t to a subsequence set $SS_t = \{sr_1, \dots, sr_k\}$ is as follows. First subsequences in SS_t are sorted ascending according to the mission saturation defined as

$$\rho_i = n * \frac{RT_i}{TL_i} \quad (11)$$

Where n is the number of targets in subsequence sr_i , RT_i is the time spent by UAV i travels along sr_i , and TL_i is the maximum travel time permitted for UAV i .

Then t is inserted to the subsequence with the smallest mission saturation. If all the constraints defined in equations(6)-(9) are satisfied, the insertion process is end. Otherwise, t is inserted to the next subsequence and the previous steps are repeated.

3.3 Fitness Function

In GA, fitness is the accordance to judge the goodness of a chromosome. Considering the characteristics of CRMPP, we set the fitness function as

$$fitness(X) = \sum_{vij} f'_{vij} x_{vij} \quad (12)$$

Where $f'_{vij} = \alpha Dis_{ij} + \beta(T_i + s_{vi} + t_{vij} + w_j) + \gamma \max\{0, (T_i + s_{vi} + t_{vij} - l_j)\} + \eta \max\{0, (T_i + s_{vi} + t_{vij} - TL_v)\}$

The fitness function for a chromosome is consisted of four parts, distance weighted by coefficient α , total consumed time weighted by coefficient β , penalty weighted by γ for tardiness and penalty weighted by η for exceeding the maximum UAV travel time. Since reconnaissance resolution constraints has been considered in the chromosome representation, the fitness function do not include component for this. For CRMPP, finding a feasible solution is more important than reducing the total distance and consumed time. Therefore, the fitness function gives higher priority to reducing tardy UAVs and UAVs exceeding the maximum travel time. If there is no violation of the constraints, the chromosome with small distance and total travel time is better. Thus the weights for the coefficients in (12) are chosen to obtain a feasible solution first and then minimize the total distance and travel time. Therefore, γ and η are much bigger than α and β .

3.4 Evolutionary Operators

Selection. We need to select parents for mating and crossover at every generation stage. In our approach, the tournament selection strategy with elite retaining model is used to generate a new population. A set of K individuals are randomly selected from the population and the individual with the smallest fitness is selected to be the parent. In our approach, the tournament set size is $K = 2$. The idea of elitism is to carry on the best individual into the next generation, which ensures that the current best solution from the overall previous populations will never deteriorate from one generation to the next.

Subsequence Crossover. For the special chromosome representation described above, the classical PMX and OX crossover operators which are usually used in combination optimization problems produced infeasible solutions. In this paper, we presents a novel crossover problem specific operator called subsequence crossover (SSX) operator.

Given two parents, $P1 = \{sr1_1, \dots, sr1_{N_v}\}$ and $P2 = \{sr2_1, \dots, sr2_{N_v}\}$, a random integer pc between 1 and N_v is generated. The subsequence $sr1_{pc}$ and

$sr2_{pc}$ are exchanged, which results in $Child1=\{sr1_1, ..., sr2_{pc}, ..., sr1_{Nv}\}$ and $Child2=\{sr2_1, ..., sr1_{pc}, ..., sr2_{Nv}\}$. Then the repeated targets are deleted from them. Since each chromosome should contain all the target numbers, the next step is to locate the best possible locations for the missing targets. For a missing target t to be inserted into a chromosome $Child=\{sr_1, ..., sr_{Nv}\}$, according to the category C_{pt} that t belongs to, the subsequences set that t could be inserted to is $SS_t = \{sr_i | i \leq pt, \text{ and } i \neq pc\}$. Then t is inserted to SS_t as above describes. SSX operator ensures that the children will satisfy the reconnaissance resolution and time feasibility constraints.

Forward Insert Mutation. Mutation aids GA to break away from fixation at any given point in the search space. While using mutation it may be better to introduce the smaller destruction on the good schemas, especially in the CRMPP, where the reconnaissance resolution constraints and time windows can easily be violated. We put forward a novel mutation operator called forward insert mutation(FIM) operator for CRMPP. A randomly selected target t belonging to subsequence sr_i currently, is inserted to the subsequences set $SS_t = \{sr_1, ..., sr_{i-1}\}$ as above described. Then t is deleted from subsequence sr_i . This FIM operator ensures that the mutated chromosome will still satisfy the reconnaissance resolution constraints and time feasibilities.

4 Simulation Results

4.1 A Straightforward Experiment

First we present a straightforward CRMPP test problem that 3 UAVs are asked to conduct reconnaissance on 20 targets distributed in a $100m \times 100m$ square area. The reconnaissance resolution of each UAV is $r_1 = 1m$, $r_2 = 10m$, $r_3 = 20m$, and maximum travel time permitted for UAV is $TL_1 = 900$, $TL_2 = 1000$, $TL_3 = 1000$.

The parameters of our GA algorithm are set as

- population size = 100
- generation span = 1000
- crossover rate = 0.80
- mutation rate = 0.30

Fig.3 a. illustrates the feasible solution of the CRMPP test problem constructed by algorithm *IFSG*. The optimal solution given by our algorithm is shown in Fig.3 b. Both the two solutions satisfy all the constraints of the problem. Table 1 gives the objectives of the two solutions. We can see that the optimal solution reduces the travel distance of all three UAVs, and the performance is increased to a great extent. Please note that due to the reconnaissance resolution and time window constraints, there are still some detours in the final solution, which different CRMPP from TSP and VRP problems.

Fig.4 shows the convergence trace of the best solution's cooperative reconnaissance(CR) cost at each generation along the evolution. We can see that the

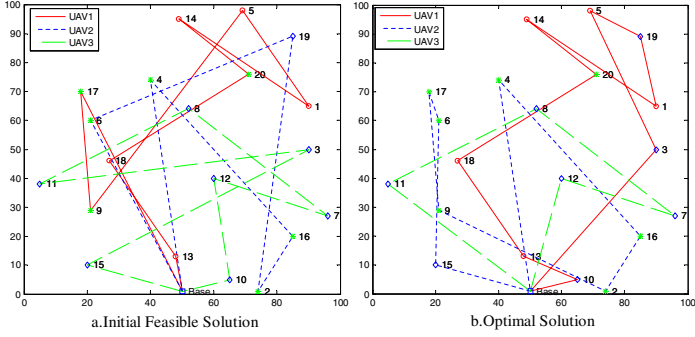


Fig. 3. Feasible solution and optimal solution of CRMPP test problem

Table 1. Comparison of the initial feasible solution and the optimal solution

	travel distance				total consumed time			
	UAV 1	UAV 2	UAV 3	total	UAV 1	UAV 2	UAV 3	total
Feasible solution	412.58	316.94	382.62	1112.14	847	916	942	2705
Optimal solution	349.55	285.07	207.74	842.36	794	943	409	2146

decline of the CR cost is faster at the earlier as compared to later. Although it is difficult to prove that we have find the optimal solution, we can strongly believe that the algorithm has optimized the objective of CRMPP effectively.

4.2 Experiments with Modified Solomon’s VRPTW Instances

In order to verify the efficiency of our algorithm, we modify the Solomon’s VRPTW instances[8] to be CRMPP instances. Generally speaking, CRMPP has wide time window for the base and the targets are usually uniformly distributed geographically. Therefore, R2 and RC2 categories of the six Solomon VRPTW instances are selected to be the seeds of our CRMPP instances. In our CRMPP instances, the locations and time windows for the targets are the same with the Solomon instances, while the number of vehicles and the reconnaissance resolution demands are changed to consist with CRMPP.

We do the simulations with 4 different instances(r201, r202, rc201 and rc202), where 8 UAVs with different capabilities are asked to conduct reconnaissance on 100 targets. Each instance is calculated 100 times and average result is given to eliminate the random effects. Table2 shows the simulation results on the 4 instances, where the CR cost of the initial feasible solutions and the optimal solutions are also given. From the table, we can see that the optimal solution can decrease the CR cost of the initial feasible solution 30% averagely in the 4 instances. From the results, we are convinced that the objective of the CRMPP has been optimized by our approach.

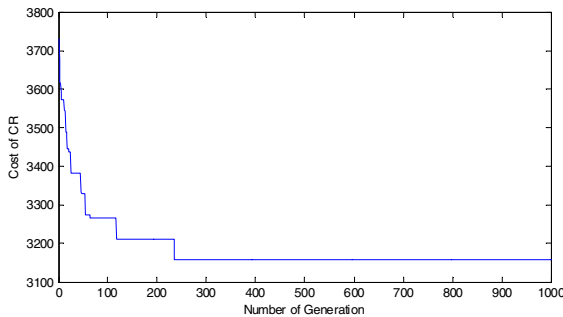


Fig. 4. Convergence trace of CRMPP test problem

Table 2. Simulation results of 4 different CRMPP instances

Prob.	feasible solution	optimal solution	CR cost decrease
r201	5137.7	3563.8	30.63%
r202	4989.3	3624.9	27.35%
rc201	4798.6	3510.2	26.85%
rc202	4863.6	3208.0	34.04%

5 Conclusions

While many previous studies ignored the targets demands for reconnaissance resolution and time windows, this paper presents a mathematical formulation taking the above factors into account. Then GA based approach is used to resolve the special NP-hard problem. Considering the characteristics of CRMPP, the integer string chromosome representation is introduced which restricts the solution to satisfy the reconnaissance resolution demands. Based on the chromosome representation, a novel subsequence crossover operator is designed to ensure the children satisfying the resolution constraints. A forward insertion mutation operator is also designed to increase the population diversity and ensure that the mutated individuals are legal. Initial population of our approach is composed of feasible solutions and random solutions, thus the algorithm will not fall into the local optimal and will not convergence slowly either. Finally, we verify the efficiency of our algorithm with extensive CRMPP instances. Simulation results show that our approach has good convergence trace and the optimal solution can increase the performance of the initial feasible solution greatly.

References

1. Ryan,Joel L., Bailey,T.G., Moore,J.T., and Carlton, W.B.: Reactive Tabu Search in Unmanned Aerial Reconnaissance Simulations. Proceedings of the 1998 Winter Simulation Conference, Grand Hyatt, Washington D.C. (1998) 873-879

2. Hutchison, M.G.: A Method for Estimating Range Requirements of Tactical Reconnaissance UAVs. AIAA's 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles, Portsmouth, Virginia (2002) 120-124
3. Ousingsawat, J., and Campbell, M.E.: Establishing Trajectories for Multi-Vehicle Reconnaissance. AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island (2004) 1-12
4. Savelsbergh, M.W.P.: Local Search for Routing Problems with Time Windows. *Annals of Operations Research* 4. (1985) 285-305
5. Thangiah, Sam R.: Vehicle Routing with Time Windows using Genetic Algorithms. in L. Chambers, (ed), *Practical Handbook of Genetic Algorithms: New Frontiers*, II, (1995) 253-277
6. Ombuki, B., Ross, B.J., and Hanshar, F.: Multi-objective Genetic Algorithms for Vehicle Routing Problem with Time Windows. Technical Report # CS-04-2. Brock University. (2004)
7. Berger, J., Barkaoui, M. and Bräysy, O.: A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows. Working paper, Defense Research Establishment Valcartier, Canada. (2001)
8. Solomon, M.M.: Algorithms for Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*. Vol.35, No.2 (1987) 254-265

Improving SVM Training by Means of NTIL When the Data Sets Are Imbalanced*

Carlos E. Vivaracho

Dep. Informática, U. de Valladolid
cevp@infor.uva.es

Abstract. This paper deals with the problem of training a discriminative classifier when the data sets are imbalanced. More specifically, this work is concerned with the problem of classify a sample as belonging, or not, to a Target Class (TC), when the number of examples from the “Non-Target Class” (NTC) is much higher than those of the TC. The effectiveness of the heuristic method called *Non Target Incremental Learning* (NTIL) in the task of extracting, from the pool of NTC representatives, the most discriminant training subset with regard to the TC, has been proved when an Artificial Neural Network is used as classifier (ISMIS 2003). In this paper the effectiveness of this method is also shown for Support Vector Machines.

1 Introduction

A set of examples (or training set) is said to be imbalanced if one of the classes is represented by a small number of cases as compared to the other classes. In this situation the inductive learning systems may have difficulties to learn the concept related with the minority class.

This situation can be found in a lot of real world applications [2][5], including biometric person recognition. Although there have been a number of attempts at dealing with that problem [2][5][14], these attempts were mostly conducted in isolation. It was so until the year 2000, when the first workshop on the topic was held (AAAI-Workshop on Learning from Imbalanced Data Sets). There has not been, to date, much systematic strive to link specific types of imbalances to the degree of inadequacy of standard classifiers, nor have there been many comparisons of the various methods proposed to remedy the problem.

These solutions to remedy the class imbalance problem have been proposed both at the data and algorithmic levels. This work is concerned with the first one.

At the data level, the different solutions can be included in the following two forms of re-sampling [2][1]:

* This work has been supported by Ministerio de Ciencia y Tecnología, Spain, under Project TIC2003-08382-C05-03.

- **over-sampling** the minority class until the classes are approximately equally represented. This over-sampling can be accomplished with random replacement, directed replacement (in which no new samples are created but the choice of samples to replace is informed rather than random) and with informed generation of new examples.
- **under-sampling** the majority class until the classes are approximately equally represented. This method can be performed randomly or directed, where the choice of examples to eliminate is informed.

Our work is concerned with biometric person recognition, where the probability distribution of the examples is unknown and the generation of artificial samples is not recommended [7]. Thus, over-sampling by means of informed generation of new examples is not a good idea. With regard to the other over-sampling technique, replacement, it has shown good results in the past [4][13] with small databases and, then, small imbalance in the data sets. With the maturing of the problem, the databases became more realistic, increasing the level of class imbalance. Under these conditions minor class sample replacement significantly increases the training time, and, besides, an over-learning of the classifier was observed, decreasing the system performance. For these reasons our efforts were focused on the under-sampling solution.

At this point, let us make the problem a bit more specific and let us fix some notation. The goal in the so called two-class classification problem is to classify a sample as belonging, or not, to a certain class called the Target Class (TC), when samples of TC and "Non-Target Class" (NTC) are both available [6]. If this task is approached by means of a discriminative classifier, it must be trained with examples from both the TC and the NTC. Then, the problem to be solved, under-sampling the major class, lies in selecting the most informative examples from a pool of representatives of the NTC, so as not to not discard potentially useful information.

In [12] a new directed sampling algorithm called *Non-Target Incremental Learning* (NTIL) was proposed. In this algorithm is the classifier which directs the most discriminative NTC examples subset selection, during the learning stage. In that work the effectiveness of NTIL when an ANN is used as classifier was shown.

In this work the same task is approached, the user identification by means of his/her voice (Speaker Verification, SV), but using a Support Vector Machine (SVM) as classifier. The SVMs have been chosen because, theoretically, this classifier represents an adverse use case of NTIL, as SVMs have proved not to be sensitive to class imbalance in the data [6]. Besides, it is a classifier that, in principle, does what NTIL sets out to do. In spite of that, we will show the good performance of NTIL with SVM, showing the effectiveness of this selection algorithm.

This paper is organized as follows. Section 2 gives a more in depth study of the NTC training examples selection problem. The heuristic solution NTIL will be shown in section 3. In section 4 will be seen the application example of NTIL when a SVM is used as classifier. The conclusions are presented in section 5.

2 The Selection Problem

To reduce notation, the NTC samples used in classifier learning will be called *Subset for Training* (ST), and the pool of examples of the NTC used for extracting the ST will be called the *Development Set* (DS).

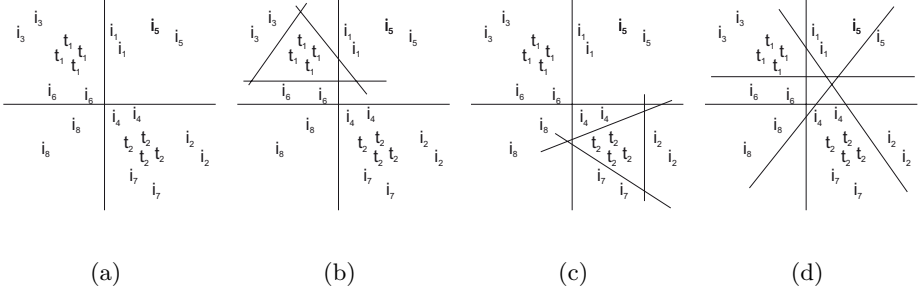


Fig. 1. (a) Example of characteristic vectors distribution. (b) Example of good ST selection to classify TC 1, by means of linear discriminative functions. (c) The same for TC 2. (d) Example of a bad ST selection for TC 2.

A simple but representative, 2-dimensional linear separable example is used to better show the influence of the ST selection on the system performance. Let us suppose that the distribution of the two-component characteristic vectors extracted from the training samples is that shown in fig. 1(a), where t_j are the vectors extracted from the training sample/s of TC j , and i_k are the vectors extracted from the DS sample I_k (i_k are, then, the NTC examples).

Let us suppose, without loss of generalization in the reasoning, that linear discriminative functions are used.

To train the TC 1 classifier, the best selection is the NTC samples I_1 , I_3 and I_6 . The linear discriminative frontiers between this TC and each of the selected ST, are shown in fig. 1(b). In this figure, it can be seen that the use of another samples does not add any information to the classification, and can impair the system performance. In the same way, to get a good TC 2 classification, it is sufficient to use the vectors from I_2 , I_4 and I_7 . In fig. 1(c) the linear discriminative frontiers between the selected training samples can be seen.

Finally, in fig. 1(d), the problems with an arbitrary random selection can be seen: if the ST selection to train the T_2 classifier was, for example, I_2 , I_5 and I_8 , the probability of wrongly accepting NTC samples would be very high.

The example, although simple, clearly shows the importance of the ST selection, in both size and composition.

3 Heuristic Choice Proposals for the ST

With a directed (heuristic) selection of the ST we try to find an under-sampling method that is:

- *Effective*: the ST must be representative of the NTC.
- *Robust* with respect to the DS. The effectiveness must be independent of this set.

The following algorithm has shown that complies with these conditions [12].

3.1 Non Target Incremental Learning Algorithm

In this proposal is the current classifier, during the learning stage, which directs the selection of the most informative ST. Thus, the use of additional systems and the necessity of modeling the training set distribution are both avoided.

The proposed selection technique is based on the idea of *Incremental Learning*, but in this case, instead of modifying the classifier size during training, modifications affect the composition of the training set. To be more precise, that corresponding to the ST. The proposed technique can be formulated through the following steps:

- a) The size of the ST (*MaxST*) and the ending criteria of the classifier training (*EndCri*) are selected.
- b) The classifier is trained using the TC training sample/s and one of those in the DS until *EndCri* is reached. We begin with a preliminary, weak classifier.
- c) The score of the classifier $S(X)$ is obtained for each of the remaining samples of the DS, and the N with the highest values¹ are chosen as the most similar (nearest) to the TC. If the classifier were used for classification, these N samples would be those with the highest probability of producing false acceptances; though, due to their discriminative capability, these are selected to train the classifier in the next phase.
- d) The N samples chosen in step c) are included into the ST set and the classifier is trained once again.
- e) Steps c) and d) are repeated until *MaxIT* is reached.

The ST selection in each iteration of the previous technique can be seen, although not in the strict sense, as the N “non yet learned” samples.

In order to avoid random initialization of the ST, even the first ST sample (step b), is also heuristically selected as is shown in [12].

4 Application Example: SVM-Based SV System

SVM have recently proved capable of providing good performance in the SV task, achieving a performance close to that of the best GMM [10]. However, our goal here is not to prove this competitiveness, but to prove the performance of NTIL. So, no comparisons with other SVM-based systems are going to be performed.

¹ We are assuming that the classifier is trained to give scores for the TC samples higher than those for the NTC samples. If not, the modification in the reasoning is immediate.

4.1 The Speaker Verification Task

Speaker Verification is a subtask of Automatic Speaker Recognition, whose goal is the validation of the claimed user identity by means of his/her voice.

The comparison can be made both for text-dependent and text-independent SV. Text-dependent SV aims to identify the speaker from the utterance of a speech sample known “a priori”. On the other hand, in text-independent SV, the speech content of the utterance is not known in advance. The experiments have been performed using this second option.

4.2 Speech Corpus

A different corpus of that used in [12] has been used. We performed experiments based upon NIST² (National Institute of Standards and Technology, USA) 2002 one-speaker detection, cellular data task. The data (available from the Linguistic Data Consortium, LDC) will be taken from the second release of LDC’s cellular switchboard corpus. The training data for a target speaker is two minutes of speech from that speaker, excerpted from a single conversation. Each test segment is the speech of a single speaker, excerpted from a one-minute segment taken from a single conversation. Evaluation trials for cellular data will include both “same number” and “different number” test. The DS used consists of 82 speakers for male target speakers (“measured” in feature vectors, the proportion between target speaker training vectors and DS vectors is 1:50) and 56 for female target speakers (the previous ratio is now 1:30).

Due to computational load requirements³, a representative subset of the target speaker has been selected for the experiments. Previous tests showed that the results achieved with the selected subset of 15 males and 15 females were not significantly different from those achieved with all the target speakers. Then, it can be said that, in spite of its size, this subset provides representative results. The test set has 306 genuine trials and 3934 impostor trials.

4.3 Parameter Extraction

Standard feature extraction is accomplished from each speech sample through 18 mel-frequency cepstral coefficients (MFCC), plus 18 Δ MFCC vectors. These are extracted from 32 ms frames, taken every 16 ms with Hamming windowing. Pre-emphasis factor of 0.97 is accomplished, and cepstral mean Subtraction (CMS) is applied as the channel compensation scheme.

4.4 System Performance Measurements

The system evaluation has been accomplished under one-speaker detection task rules of the NIST speaker Recognition Evaluation Plan.

² The NIST evaluations are the standard and independent ones most important in SV.

³ The time necessary to train an SVM in experiment 3 (see Sec. 4.8), **where NTIL is not used**, is measured in days, even using a very new computer.

The system performance measure can be graphically performed by DET (Detection Error Tradeoff) curves [8], where the False Match Rate (FMR) or False Alarm Probability (the rate of NTC samples falsely accepted) is plotted against the False Non-Match Rate (FNMR) or Miss Probability (the rate of TC samples falsely rejected), for several decision thresholds, and with gaussian axes. An example can be seen in the figure 2. Is the measure recommended by NIST and other important biometric groups.

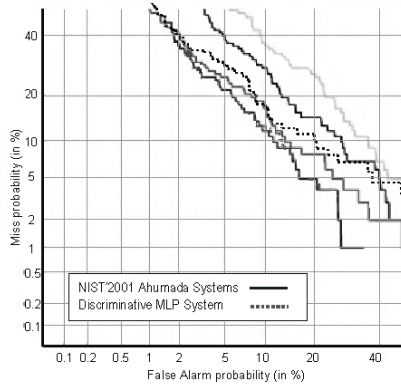


Fig. 2. Performance comparison, by means of a DET curve, between the discriminative MLP-based system shown in [12] and the best ones in NIST 2001 speaker verification evaluation, the last NIST evaluation that included test with AHUMADA sub-corpus

However, if the number of comparison is high the use of a single number measure is more useful and easy to understand. One of the most popular is the Equal Error Rate (EER) that is the error of the system when the decision threshold is such that the FMR equals the FNMR: in the DET curve, the point where the diagonal cuts the curve. The EER will be the measure used in this paper.

4.5 The System

An SVM is trained per target speaker, using the C-Support Vector Classification [3][11] included in the library for SVM LIBSVM. The popular radial basis function has been used as the kernel function.

The model can be trained only for classification (the model output is the class prediction) or for probability estimate. Both options have been tested. The performance of the system was much better when the second option was used. Besides, the degradation of this performance with the level of imbalance in the data was higher using the first option. Thus, only the results achieved when the SVM is trained for probability estimate will be shown.

Given an input vector x_i from a test (speech) sample $X = \{x_1, x_2, \dots, x_M\}$, the output of the λ_C SVM, trained to verify the C target speaker identity, is the $P(\lambda_C/x_i)$. Then, the probability that a test sample X belongs to the speaker C will be:

$$P(\lambda_C/X) = \prod_{i=1}^M P(\lambda_C/x_i) \quad (1)$$

This value can be very low, so, to avoid the loss of accuracy in its codification, the use of the logarithm is advisable:

$$\log(P(\lambda_C/X)) = \sum_{i=1}^M \log(P(\lambda_C/x_i)) \quad (2)$$

The result in eq. 2 is highly dependent on M . Additional use of the mean to avoid this dependence also allows the result to be bounded. Thus, the final score, $S(X)$, of the system per test sample X will be:

$$S(X) = \frac{1}{M} \sum_{i=1}^M \log(P(\lambda_{S_c}/x_i)) \quad (3)$$

To deal with the class imbalance problem LIBSVM provides an algorithm level solution [9], using different penalty coefficients for each class to be learnt. As can be seen, the solution is effective, but substantially increases the learning time and the model size. The performance of NTIL has been tested with and without that modification in the SVM learning algorithm.

4.6 Experiments

In order to show the NTIL performance, each target speaker model (SVM) is trained as follows:

- **Experiment 1.** Under-sampling the major class by means of NTIL. The standard learning algorithm was used to train the SVM.
- **Experiment 2.** Without under-sampling and using the standard SVM learning algorithm. That is, the SVM is trained with all the examples in the DS.
- **Experiment 3.** Without under-sampling and using the modified SVM learning algorithm to deal with the imbalance in the data sets. As before, the SVM is trained with all the examples in the DS, but each class is penalized inversely to its examples proportion in the data sets: penalty 1 for NTC and penalty 50 for TC if the speaker is male and 30 if the speaker is female.

4.7 System Evaluation

In order to get an easy comparison, the EER will be used as the system performance measure. This EER is calculated for each target speaker, the final system

performance measure being the average of these individual EER (column EER in table 1).

In order to get a more complete comparison, the time necessary to train (generate) a TC model and the size of the model (measured in number of support vectors and size of the model file created by LIBSVM) in each experiment has been calculated.

4.8 Results

The results of the experiments can be seen in table 1. The model training time is shown in minutes, and the model file size in MBytes. The computer used in the experiments is a PC with an Opteron (AMD) processor at 2.6 GHz. Different computers would give different times, so the most important is not the absolute training time, but the relative difference. When NTIL is used the “training time” includes all the necessary operations to build the model: first ST sample selection, weak classifier training and n repetitions of the c and d steps. All the system performance measures for each NTIL algorithm iteration (“NTIL It i ” in the table) have been obtained. The value of N in the NTIL algorithm has been fixed to get, approximately, the same number of vectors as those in the TC training set in each iteration. That is, if, for example, the number of target speaker C training vectors is 5000, we will choose, in each NTIL iteration, the N samples of the DS with the highest scores such that the sum of the vectors of these N samples will be approximately equal to 5000.

Table 1. Final results of the experiments with SVM. *EER* is the mean of the EER calculated for each target speaker in %. *Training time* is the time needed to build the speaker model, measured in minutes. *nSV* is the number of Support Vectors used. *file size* is the size of the model file created by LIBSVM in MBytes.

	EER(%)	Model Size		
		training time	nSV	File Size
Exp. 1, NTIL It. 1	16.3	16	11459	5
Exp. 1, NTIL It. 2	11.8	44	11685	5.1
Exp. 1, NTIL It. 3	11.4	78	11862	5.2
Exp. 1, NTIL It. 4	11.7	117	11930	5.2
Exp. 2	14.3	255	12228	5.4
Exp. 3	12.7	2820	141448	62

From the results in the table, it can be seen that the use of NTIL improves the SVM performance, since the results are better than those achieved without using NTIL, from the second iteration of the algorithm. In order to compare approaches, three NTIL iterations will be used as experiment 1 system configuration, as the best results have been achieved with this. With regard to the standard SVM (experiment 2), the NTIL-based system improves the EER by 20%,

with a reduction in the training time of 70%, while the model size is similar. Comparing our solution to deal with class imbalance with that in LIBSVM, NTIL achieves 10% more in the EER reduction, while the model size is 12 times less, and the training time 36 times less.

5 Conclusions

In previous works the performance of the heuristic NTIL technique to select the most discriminant ST and to avoid unpredictable behavior of the system when an ANN is used as classifier was shown. In this work the effectiveness of NTIL has been proved again, but here when a SVM is used, in spite of that, theoretically, this classifier is less sensitive to class imbalance in the data. In [12] the effectiveness and the robustness of NTIL with regard to the DS was shown, in this work the robustness of NTIL with regard to the classifier has been proved.

References

1. Gustavo Batista. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD explorations*, 6(1):20–29, 2004.
2. Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial: Special issue on learning from imbalance data sets. *SIGKDD explorations*, 6(1):1–6, 2004.
3. C. Cortes and V. Vapnik. Support-vector network. *Machine Learning*, (20):273–297, 1995.
4. Assaleh K. T. Farrel K. R., Mammone R. J. Speaker recognition using neural networks and conventional classifiers. *IEEE Transactions on Speech and Audio Processing*, 2, No. 1, part II, 1994.
5. Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A sistematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.
6. Piotr Juszczak and Robert P.W. Duin. Uncertainty sampling methods for one-class classifiers. In *Proc. of the Workshop on Learning from Imbalanced Datasets II, ICML*, 2003.
7. A. J. Mansfield and J. L. Wayman. Best pratices in testing and reporting performance of biometric devices. version 2.01. Technical report, 2002.
8. B. Martin del Brio and A. Sanz Molina. *Redes Neuronales y Sistemas Borrosos*. Ra-Ma, 1997.
9. E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. Technical report, 1997.
10. Alex Solomonoff, Carl Quillen, and William M. Campbell. Channel compensation for svm speaker recognition. In *Proc. Odyssey 2004, the Speaker and Language Recognition Workshop*, 31 May-3 June 2004.
11. V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
12. Carlos E. Vivaracho, Javier Ortega-Garcia, Luis Alonso, and Quiliano I. Moro. Extracting the most discriminant subset from a pool of candidates to optimize discriminant classifier training. *Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence, Foundations of Intelligent Systems, 14th Iternational Symposium ISMIS*, (2871):640–645, 2003.

13. C. Vivaracho-Pascual, J. Ortega-Garcia, L. Alonso-Romero, and Q. Moro-Sancho. A comparative study of mlp-based artificial neural networks in text-independent speaker verification against gmm-based systems. In Borge Lindberg Paul Dalsgaard and Henrik Benner, editors, *Proc. of Eurospeech01*, volume 3, pages 1753–1756. ISCA, 3-7 September 2001.
14. Gary M. Weiss. Mining with rarity: A unifying framework. *SIGKDD explorations*, 6(1):7–19, 2004.

Evolutionary Induction of Cost-Sensitive Decision Trees

Marek Krętowski and Marek Grzes

Faculty of Computer Science, Białystok Technical University
Wiejska 45a, 15-351 Białystok, Poland
{mkret, marekg}@ii.pb.bialystok.pl

Abstract. In the paper, a new method for cost-sensitive learning of decision trees is proposed. Our approach consists in extending the existing evolutionary algorithm (EA) for global induction of decision trees. In contrast to the classical top-down methods, our system searches for the whole tree at the moment. We propose a new fitness function which allows the algorithm to minimize expected cost of classification defined as a sum of misclassification cost and cost of the tests. The remaining components of EA i.e. the representation of solutions and the specialized genetic search operators are not changed. The proposed method is experimentally validated and preliminary results show that the global approach is able to effectively induce cost-sensitive decision trees.

1 Introduction

In many data-mining applications, especially in medicine or business, the traditional minimization of classification errors is not the most adequate scenario. Particular decisions have often significantly different impact on the overall result. It is especially evident in medical diagnosis, where misclassifying an ill person as a healthy one is generally much more dangerous (and costly) than an inverse error. In such situations different misclassification costs associated with decisions are used to compensate the problem. Additionally, the cost of decision making (i.e. the cost of used features) can be also taken into account. This is usually obtained by preferring such a classifier which gives equally accurate predictions with a lower cost, calculated as a sum of costs of the performed tests. In medical domain, similar diagnostic accuracy can be sometimes obtained with very expensive tests as well as with simple and cheap examinations.

Cost-sensitive classification is the term which encompasses all types of learning where cost is considered [12]. However most of existing cost-sensitive systems consider only one cost type. There are two main approaches to making a classifier cost sensitive. In the first group classical classifiers are converted into cost-sensitive ones. In the context of decision tree learning it encompasses mainly changing the splitting criteria and/or adopting pruning techniques for incorporating misclassification cost (e.g. [1,2,4]) or cost of tests (e.g. in *EG2* [9]). Another method of misclassification cost minimization is proposed in [10], where instance-weighting is applied, but the method requires the conversion of the cost matrix into the cost vector. Among systems, which take into account both cost

types the two approaches should be mentioned: *Inexpensive Classification with Expensive Tests - ICET* [11] and *Internal Node Strategy - INS* [7,14]. *ICET* uses the standard genetic algorithm to evolve a population of biases for modified C4.5 and *INS* is based on a *total cost* criterion for nominal attributes and two-class problems. The second group includes general methods for making an arbitrary classifier cost-sensitive. *MetaCost* [3] is based on wrapping a meta-learning stage around the error-based classifier. Another method proposed by Zadrozny *et al.* [13] uses cost-proportionate rejection sampling and ensemble aggregation. However, both mentioned approaches incorporate only misclassification costs.

In the paper, a cost-sensitive extension of our EA-based system [6] for decision tree learning is presented. In contrast to the typical top-down induction, the global method searches simultaneously for both the optimal tree structure and all tests in internal nodes. Such an approach is computationally more complex but it often allows avoiding sub-optimal solutions imposed by greedy techniques. As a result accurate and more compact classifiers are obtained. It should be underlined that necessary adaptation of our error-based system to incorporate both misclassification and test costs are limited almost only to the fitness function. Apart from slightly modified method for assigning the class labels to leaves all remaining elements of the original algorithm do not have to be changed.

The rest of the paper is organized as follows. In the next section, an evolutionary algorithm for global induction of decision trees is briefly described. In section 4 the cost-sensitive extension of our system is proposed. Preliminary experimental validation of the proposed approach is presented in section 5. The paper is concluded in the last section.

2 Global Induction of Decision Trees

In this section, main ideas of our EA-based system called *Global Decision Tree (GDT)* are briefly presented. For more detailed description please refer to [6].

Representation and Initialization. Decision trees are represented in their actual form as univariate trees where any test in an internal node concerns only one attribute. In case of a nominal attribute at least one value is associated with each branch (inner disjunction). For a continuous-valued feature typical inequality tests with boundary thresholds as potential splits are considered. A boundary threshold for the given attribute is defined as a midpoint between the successive pair of examples from different classes in the sequence sorted by the increasing value of the attribute. All boundary thresholds are calculated before starting the evolutionary induction.

Individuals in the initial population are generated using the classical top-down algorithm, but tests are chosen in a dipolar-like way [5].

Genetic Operators. There are two specialized genetic operators corresponding to classical mutation and crossover. When crossover is applied the randomly chosen parts (i.e. sub-trees or only tests) of two individuals are swapped. There are a few variants of this exchange.

Mutation-like operator is a more complex operator and is applied to a given tree node. Possible modifications depend on the node type (i.e. whether it is a leaf node or an internal node). For a non-terminal node a few possibilities exist: *i*) a completely new test can be drawn, *ii*) existing test can be altered by shifting the splitting threshold (continuous-valued feature) or re-grouping feature values (nominal feature), *iii*) the test can be replaced by another test from the tree and finally *iv*) the node can be transformed into a leaf. Modifying a leaf node makes sense only if it contains feature vectors from different classes. The leaf is transformed into an internal node and a new test is randomly chosen. The search for effective tests can be recursively repeated for all descendants.

The application of any genetic operator can result in a necessity of relocation of the input vectors between parts of the tree rooted in the modified node. Additionally local maximization of the fitness function is performed by pruning lower parts of the subtree on condition it improves the value of the fitness.

Error-Based Fitness Function. The goal of any classification system is correct prediction of class labels of new objects, however such a target function cannot be directly defined. Instead the accuracy on the training data is often used, but their direct optimization leads to the over-fitting problem. In classical systems this problem is usually mitigated by post-pruning techniques. In our approach a complexity term is introduced into the fitness function preventing the over-specialization. The fitness function, which is maximized, has the following form:

$$Fitness(T) = Q_{Reclass}(T) - \alpha \cdot (S(T) - 1), \quad (1)$$

where $Q_{Reclass}(T)$ is the re-classification quality, $S(T)$ is the size of the tree T expressed as the number of nodes and α is a relative importance of the complexity term (default value is 0.005) and a user supplied parameter.

3 Cost-Sensitive Extension

There are only two modifications (new fitness function and class label assignment) necessary for incorporating misclassification and test costs into *GDT*.

Preliminaries. Let learning set $E = \{e_1, e_2, \dots, e_M\}$ consist of M examples. Each example $e \in E$ is described by N attributes (features) $A_1, A_2, \dots, A_N$ and the corresponding feature costs are denoted by $C_1, C_2, \dots, C_N$ respectively. Additionally, a decision (class label) associated with an object e is represented by $d(e) \in D$. The set of all examples with the same decision $d_k \in D$ is denoted by $D_k = \{e \in E : d(e) = d_k\}$ and the class assigned (predicted) by the tree T to the example e is denoted by $T(e)$. Let $Cost(d_i, d_j) \geq 0$ be the cost of misclassifying an object from the class d_j as belonging to the class d_i . We assume that the cost of the correct decision is equal zero i.e., $Cost(d_i, d_i) = 0$ for all d_i .

Cost-Sensitive Fitness Function. Performance of an error-based tree is judged by the classification accuracy whereas a cost-sensitive tree is assessed by the average cost, which is a sum of the average misclassification cost and

the average test costs. This suggests replacing in the fitness function the re-classification accuracy with the expected cost of classification. The expected cost in general is not limited, but for a given dataset, cost matrix and attribute costs, the maximum misclassification and test costs can be calculated. As a result the normalized cost can be easily obtained and fits into $[0, 1]$ range.

First, the misclassification cost $MCost(T)$ of the tree T is estimated on E :

$$MCost(T) = \frac{1}{M} \cdot \sum_{e \in E} Cost(T(e), d(e)). \quad (2)$$

The maximal misclassification cost for a given dataset and a cost matrix is equal:

$$MaxMC = \frac{1}{M} \cdot \sum_{d_k \in D} |D_k| \cdot \max_{i \neq k} Cost(d_i, d_k). \quad (3)$$

By dividing $MCost(T)$ by $MaxMC$ one can obtain the *normalized misclassification cost*, which is equal 0 for the perfect classification and 1 in the worst possible case.

Similar approach can be applied to calculate *normalized average tests cost* of tree T induced from training data E . Let $A(e)$ denote the set of all attributes used in tests (on the path from the root node to the terminal leaf reached by e) necessary to classify an object e . Hence, tests cost $TC(e)$ of classifying an object e by tree T is equal:

$$TC(e) = \sum_{A_i \in A(e)} C_i. \quad (4)$$

It should be noted that renewed use of any feature in the following tests does not increase $TC(e)$. The average test cost $TCost(T)$ can be defined as follows:

$$TCost(T) = \frac{1}{M} \cdot \sum_{e \in E} TC(e). \quad (5)$$

The maximal value of this cost for the given learning set is equal:

$$MaxTC = \frac{1}{M} \cdot \sum_{i=1}^N C_i, \quad (6)$$

where all features $A_i \in A$ are necessary for making prediction for any object. Normalized average test cost is calculated by dividing $TCost(T)$ by $MaxTC$. It is equal zero for the tree composed only of one leaf and is equal 1 when in every path of the tree all features are used in tests.

Finally, the fitness function, which is minimized, is defined as follows:

$$Fitness(T) = \frac{MCost(T) + TCost(T)}{MaxMC + MaxTC} + \alpha \cdot (S(T) - 1). \quad (7)$$

Class Labels for Leaves. In standard error-based decision trees the class labels are assigned to leaves by using the majority rule based on training objects which reached a leaf-node. In cost-sensitive case the class labels for leaves are chosen to minimize the misclassification cost in each leaf.

4 Experimental Results

In this section *GDT* is compared with nonnative implementations of *ICET*, *INS* and *EG2* on four datasets that were investigated in [11]. Costs of attributes that are provided for these datasets were used and cost matrices were prepared in the following way. All values of the misclassification cost are at least equal to the sum of features cost. In the subsequent experiments presented in Table 1, penalty for misclassifying less frequent class had been increasing. Ratios of such costs were: 1.5, 2, 3 and F_{LF}/F_{MF} where F_{LF} and F_{MF} are frequencies of less and more frequent class respectively.

It can be observed that *GDT* is significantly better on *heart* data both in terms of the tree size and the average cost on the test data. This dataset is a good example of the situation when the top-down heuristic is trapped into a local minimum and the method working in a global manner is able to find more optimal structure of the tree. Another dataset in which top-down algorithms found also relatively big trees is the *pima* dataset. In this case *GDT* finds smaller trees but without improvement in terms of cost, which means that the system was able to eliminate repeated tests on previously used features. As for *bupa* and *hepatitis* datasets the results of all the algorithms except *ICET* are comparable.

Evolutionary algorithms are considered to be as good as its fitness function. These statement is in some way observed in *bupa* dataset. In this case *GDT* found very small trees which are identical in each run (standard deviation is zero). It shows that the algorithm is able to find small and relatively good result that is very stable. But on the other hand this result is not optimal. Comparisons with other algorithms show that there is still place for improvement (e.g. by tuning

Table 1. The results with different cost matrices

Dataset		<i>GDT</i>		" <i>ICET</i> "		" <i>EG2</i> "		" <i>INC</i> "	
		Size	Cost	Size	Cost	Size	Cost	Size	Cost
bupa	40/55	3.0	18.96±0.0	52.7	47.37±2.2	4	18.73	4	18.74
	40/60	3.0	19.13±0.0	52.6	48.42±2.3	5	19.56	5	19.56
	40/80	3.0	19.83±0.0	53.2	51.50±2.5	3	19.82	3	19.83
	40/120	3.0	21.22±0.0	52.1	61.86±3.3	1	22.96	1	22.96
heart	600/704	7.9	142.61±7.6	48.9	306.11±19.4	16	151.94	17	165.12
	600/900	5.8	163.25±5.5	44.8	371.58±49.8	20	187.79	20	196.82
	600/1200	5.6	199.27±27.1	46.4	382.10±40.7	24	199.56	20	227.13
	600/1800	4.1	257.9±8.1	48.3	478.74±41.9	20	239.67	20	239.67
hepatitis	50/262	6.8	30.83±4.1	5.8	34.02±4.3	6	32.11	6	30.9
	50/75	7.5	13.15±3.1	5.8	18.53±0.8	5	10.05	5	10.06
	50/100	7.3	15.34±3.2	5.7	21.98±3.1	7	7.77	5	12.84
	50/150	7.1	20.42±3.7	5.8	25.09±3.1	6	27.51	5	18.39
pima	50/92	4.3	19.92±0.6	83.9	22.86±0.7	38	18.21	24	20.05
	50/75	3.4	17.58±0.2	84.5	20.63±0.9	24	16.61	28	17.94
	50/100	4.6	21.23±0.4	83.9	24.4±1.0	31	19.18	26	22.5
	50/150	5.0	23.77±0.5	84.9	31.59±1.1	30	23.88	24	24.41

user specified α parameter). Further experiments showed that it is possible to obtain better results (e.g. for $\alpha = 0.001$ on *bupa* 40/40 we gain the best result 17.03). It means that fitness function can be further investigated and it might improve the overall performance.

5 Conclusion

In this paper, our global method for decision tree induction is extended to handle two types of cost: misclassification cost and cost of tests. The necessary modifications of the evolutionary algorithm encompass mainly the new fitness function. Even preliminary results of experimental validation show that our system is able to generate competitive cost-sensitive classifiers.

Acknowledgments. This work was supported by the grant W/WI/5/05 from Białystok Technical University. The authors thank M. Czołombitko for providing us with implementation of other cost-sensitive classifiers.

References

1. Bradford, J., Kunz, C., Kohavi, R., Brunk, C., Brodley, C.E.: Pruning decision trees with misclassification costs. In *Proc. of ECML'98*. Springer (1998) 131–136.
2. Breiman, L., Friedman, J., Olshen, R., Stone C.: *Classification and Regression Trees*. Wadsworth Int. Group (1984).
3. Domingos, P.: MetaCost: A general method for making classifiers cost-sensitive. In *Proc. of KDD'99*, ACM Press (1999) 155–164.
4. Knoll U., Nakhaeizadeh G., Tausend B.: Cost-sensitive pruning of decision trees. In *Proc. of ECML'94*, Springer LNCS 784 (1994) 383–386.
5. Krętowski, M.: An evolutionary algorithm for oblique decision tree induction, In: *Proc. of ICAISC'04*, Springer LNCS 3070, (2004) 432–437.
6. Krętowski, M., Grześ, M.: Global learning of decision trees by an evolutionary algorithm, In: *Information Processing and Security Sys.*, Springer, (2005) 401–410.
7. Ling, C., Yang, Q., Wang, J., Zhang, S.: Decision trees with minimal costs, In: *Proc. of ICML'04*, ACM Press (2004), Article No. 69.
8. Margineantu, D., Dietterich, T.: Bootstrap methods for the cost-sensitive evaluation of classifiers, In *Proc. of ICML'2000*, Morgan Kaufmann (2000) 583–590.
9. Nunez, M.: The use of background knowledge in decision tree induction, *Machine Learning* **6**, (1991) 231–250.
10. Ting, K.: An instance-weighting method to induce cost-sensitive trees. *IEEE TKDE* **14**(3), (2002) 659–665.
11. Turney, P.: Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *J. of Artif. Intel. Res.* **2**, (1995) 369–409.
12. Turney, P.: Types of cost in inductive concept learning. In *Proc. of ICML'2000 Workshop on Cost-Sensitive Learning*. Stanford, CA (2000).
13. Zadrozny, B., Langford, J., Abe. N.: Cost-sensitive learning by cost-proportionate example weighting concept learning. In *Proc. of ICDM'03*. IEEE Press (2003).
14. Zhang S., Qin, Z., Ling, C. Sheng, S.: Missing is useful: Missing values in cost-sensitive decision trees. *IEEE TKDE* **17**(12), (2005) 1689–1693.

Triangulation of Bayesian Networks Using an Adaptive Genetic Algorithm

Hao Wang¹, Kui Yu¹, Xindong Wu^{1,2}, and Hongliang Yao¹

¹ Department of Computer Science and Technology, Hefei University of Technology,
Hefei 230009, China
jsjxwangh@hfut.edu.cn

² Department of Computer Science, University of Vermont, Burlington, VT 05405, USA
xwu@cems.uvm.edu

Abstract. The search for an optimal node elimination sequence for the triangulation of Bayesian networks is an NP-hard problem. In this paper, a new method, called the TAGA algorithm, is proposed to search for the optimal node elimination sequence. TAGA adjusts the probabilities of crossover and mutation operators by itself, and provides an adaptive ranking-based selection operator that adjusts the pressure of selection according to the evolution of the population. Therefore the algorithm not only maintains the diversity of the population and avoids premature convergence, but also improves on-line and off-line performances. Experimental results show that the TAGA algorithm outperforms a simple genetic algorithm, an existing adaptive genetic algorithm, and simulated annealing on three Bayesian networks.

1 Introduction

Bayesian networks ^[1] (BNs) are an important tool for knowledge representation and probabilistic inference under uncertainty. However, both exact and approximate inferences on Bayesian networks are NP-hard problems ^[2]. An effective reasoning method on a Bayesian network transforms the Bayesian network into a secondary graphical structure called a junction tree ^[3] and performs inference by a message passing protocol.

The complexity of reasoning on a junction tree is determined by the largest cliques of the junction tree. Therefore, the key point in reducing the complexity is to reduce the sizes of the largest cliques as much as possible. The sizes of the cliques of a junction tree are directly determined by the triangulation method during the construction of the junction tree. Moreover, the optimal triangulation is completely determined by the order in which the nodes are eliminated in the BN's moral graph. Hence, the search for the optimal triangulation is equivalent to the search for the optimal node elimination sequence.

The main studies of triangulation methods of moral graphs include the lexicographic search (Rose, 1976)^[4], the maximum cardinality search (Tarjan and Yannakakis, 1984)^[5], the minimum weight heuristic (Kjærulff, 1990)^[6], simulated annealing (Kjærulff, 1992)^[7], and triangulation with genetic algorithms (Larranaga, 1997)^{[8][9]}. The methods in [4] and [5] add filling edges between non-adjacent vertices in the

adjacency set of a vertex that is being eliminated, but this produces redundant edges. Hence, it is difficult to use those methods to find the minimal junction tree. The minimum weight heuristic in [6] adds recursive thinning algorithms based on the lexicographic search. It deletes redundant edges produced by the two methods in [4] and [5] and improves the search performance.

A genetic algorithm was used to search for the optimal node elimination sequence by Larranaga^[8] and the results were as good as those of simulated annealing^[7]. Unfortunately, the genetic algorithm has several disadvantages that are difficult to overcome, as a standard genetic algorithm has (1) premature convergence, (2) a poor capability for local searching, and (3) the problem of local optima. Simulated annealing has a strong capability for local searching, but it is not suitable for global searching, which makes it difficult for the search process to find the most promising basin and get the global optima.

In this paper, a new triangulation algorithm, called TAGA (Triangulation using an Adaptive Genetic Algorithm), is proposed based on an adaptive genetic algorithm^[10] to find an optimal node elimination sequence. The TAGA algorithm adopts an adaptive rank-based operator that can dynamically adjust the selection pressure of the population at different stages of evolution. That is, it uses a smaller selection pressure in the early and middle stages of evolution. TAGA accepts some worse solutions in favor of maintaining the diversity of the population and accomplishing a global search. To use a larger selection pressure during the last stage, TAGA no longer accepts any worse solutions and it finds an optimal solution from the current solutions as soon as possible. Hence, TAGA brings a linear rank-based operator^[11] into adaptive crossover and mutation operators naturally, and improves both on-line and off-line performances. Meanwhile, it maintains the diversity of the population, avoids premature convergence, and enhances its capability for global searching. In order to prevent the best solution from being destroyed, we adopt the elitist strategy^[12].

The rest of the paper is organized as follows. Section 2 introduces the background knowledge. In Section 3 the TAGA algorithm is presented, and in Section 4 we provide experimental results with an analysis.

2 Background

The following are relevant terms that are used in this paper.

Definition 1. Moral graph The moralization of a Bayesian network structure means all variables with a common child are linked, after which the directions on the arcs are deleted

Definition 2. Triangulated graph A graph is triangulated iff all cycles of length four or more have a chord. A chord of a cycle is a connected pair of nodes that are not consecutive in the cycle.

Definition 3. Potential A potential over a set of variables X is a function that maps each instantiation x into a nonnegative real number. This potential is denoted by ϕ_x .

Definition 4. Cliques A clique in an undirected graph G is a subgraph of G that is complete and maximal. Being complete means that every pair of distinct nodes is

connected by an edge. Being maximal means that the clique is not properly contained in another larger, complete subgraph. Each clique C has an associated set of variables V_C and a clique potential ϕ_C . ϕ_C is defined as

$$\phi_C = \prod_{X \in V_C} P(X \mid pa(X)) \quad (1)$$

where $pa(X)$ represents the parents of X . The potential functions of empty cliques are a single unit.

Definition 5.^[3] **Junction Trees** A junction tree is represented as $T=(\Gamma, \Delta)$, where Γ is the set of clusters, and clusters in Δ is used to connect two clusters in Γ . For a pair of adjacent clusters C_i and C_j , where $C_i \in \Gamma, C_j \in \Gamma$ and $S_k \in \Delta$, S_k is a separator between C_i and C_j . $S_k = C_i \cap C_j$

Definition 6.^[3] **Weight of a junction tree** A clique $C=\{v_1, v_2, \dots, v_k\}$ is generated from a triangulated graph as it is being constructed. The node v_i is one of the nodes of the clique C ; k represents the number of nodes of C ; $w(v_i)$ is the number of values of v_i ; and the weight of the clique C is defined as follows:

$$w(C) = \prod_{i=1}^k w(v_i) \quad (2)$$

The weight of a junction tree is defined in Eq (4).

$$w(JT) = \log_2 \sum_C w(C) \quad (3)$$

The basic idea of transforming a Bayesian network into a junction tree is to perform probabilistic inference on the tree by a message passing protocol.

2.1 Junction Tree Construction

In this section, we summarize how to construct a junction tree from a Bayesian network. A series of graphical transformations that result in a joint tree can be summarized as follows:

Step 1. Construct an undirected graph, called a moral graph, from the directed acyclic graph (DAG) representation of the Bayesian network.

Step 2. Selectively add arcs to the moral graph to form a triangulated graph.

Step 3. From the triangulated graph, identify selected subsets of nodes, called cliques.

Step 4. Build a junction tree: connect the cliques to form an undirected tree that satisfies junction tree properties, and insert appropriate separators.

Steps 2 and 4 are nondeterministic. Consequently, many different junction trees can be built from the same DAG of a Bayesian network. Forming a triangulated graph is the most important step. According to Eqs. (1), the cliques generated from Step 2 directly determine the complexity of inference on the junction tree.

2.2 Triangulation of a Bayesian Network

The triangulation of a Bayesian network can use the theorem of elimination proposed by Rose^[4], and the process is summarized as follows.

Firstly, according to the moral graph, an elimination ordering of the vertices (which specifies the order in which these vertices should be eliminated) is established. Let δ represent the elimination ordering, namely, $\delta = \{1, 2, \dots, |V|\}$. δ maps the nodes of V into the elements of $\{1, 2, \dots, |V|\}$ where $|V|$ is the base of the set of variables in V . Secondly, according to the elimination ordering, the vertices of the moral graph are eliminated one by one. A vertex v is eliminated by adding edges such that the vertices adjacent to v are pairwise adjacent after the deletions of v and its incident edges. Hence the cliques of the junction tree result from eliminating the vertices.

Since the moral graph is fixed, the elimination ordering of the moral graph determines the triangulated graph and its cliques exclusively, and the junction tree is determined at the same time.

A simple random elimination in which the vertices are selected at random cannot guarantee to produce the minimal triangulations either in terms of the number of filling edges or in terms of the size of the state space. Consequently how to search for an effective elimination ordering of the vertices (which makes the weight of the triangulated graph as small as possible) is the key problem. This problem has been proven to be NP-complete^[13].

The cliques in the junction tree shall have joint probability tables attached to them. The size of each table is the product of the numbers of states of relevant variables. The total size increases exponentially with the sizes of the cliques. A good triangulation, therefore, is a triangulation that yields small cliques; or to be more precise, yields small probability tables. According to Eq (3), the weight of the junction tree reflects more precisely the probability tables, and also reflects the complexity of reasoning of the junction tree. Therefore, according to the optimization criteria proposed in [4][5][6], we choose the minimal weight of the junction tree as the optimization criterion for our elimination ordering.

3 The TAGA Algorithm

3.1 Encoding Mechanism

The problem of searching for the optimal elimination ordering of the vertices is similar to the well-known Traveling Salesman problem, both of which belong to typical combinatorial optimization problems. In the Traveling Salesman problem, path representation is the most natural and easiest way to express gene coding that corresponds to the route. In this paper, path representation is used to code the problem of node elimination ordering, in the form of an array. That is, if a node elimination is put in an order like $\{v_2, v_4, v_6, v_9, v_{10}, v_7, v_1, v_3, v_5, v_8\}$, then $\{2, 4, 6, 9, 10, 7, 1, 3, 5, 8\}$ is used to represent the encoding mechanism.

3.2 The Fitness Function

The weight of the junction tree reflects the size of the probability tables, and also the complexity of reasoning on the junction tree. Therefore, the weight function of the junction tree is the most suitable choice to measure the fitness function.

3.3 Adaptive Crossover and Mutation Operators

In simple genetic algorithms, crossover and mutation probabilities are fixed. Therefore, the convergence speed of these genetic algorithms is slow, and they are prone to finding local optima. An adaptive genetic algorithm (AGA)^[10] adjusts crossover and mutation probabilities according to the fitness of chromosomes. The detailed adjusting method is as follows.

$$P_c = \begin{cases} k_1(f_{\max} - f') / (f_{\max} - f_{\text{avg}}), & f' > f_{\text{avg}} \\ k_3 & f' \leq f_{\text{avg}} \end{cases} \quad (4)$$

$$P_m = \begin{cases} k_2(f_{\max} - f) / (f_{\max} - f_{\text{avg}}), & f > f_{\text{avg}} \\ k_4 & f \leq f_{\text{avg}} \end{cases} \quad (5)$$

Where P_c represents the crossover probability, P_m represents the mutation probability, $f_{\max}$ denotes the maximal fitness value of the population, f_{avg} is the average fitness value of the population, f' indicates the larger fitness value of the two crossover solutions, and f represents the fitness value of the solution that is going to mutate. Here, k_1, k_2, k_3, k_4 are constants between 0 and 1, and k_1 and k_3 are bigger than the other two constants.

With the evolution of the population, AGA adaptively adjusts crossover and mutation probabilities. Unfortunately, this mechanism is only suitable for the population in the later stages of its evolution. It would slow down the evolution process in the early stage, because the better solutions of the population always gain the dominance in the early stage, which keeps the evolution on the local optima. To deal with this problem, an adaptive rank-based selection operator is proposed in Section 3.4.

3.4 Adaptive Linear Rank-Based Selection Operator

A selection operator always prefers to select superior solutions of the population, and the selection pressure determines the expected number of superior solutions to survive in the next generation of the population. The larger the selection pressure is given, the larger the expected number of superior solutions will survive. The convergence of genetic algorithms depends on the selection pressure of the population. If the selection pressure is too small, the speed of convergence will be very slow. In contrast, if the selection pressure is too huge, it will be easy for genetic algorithms to converge to the local optima. Moreover, finding a proper selection operator to maintain the diversity of the population also helps to avoid premature convergence.

Because of the above-mentioned drawback of adaptive genetic algorithms, we propose an improved linear rank-based selection operator to make sure that the selection

pressure of the population will be adjusted dynamically, in order to improve the performance of the adaptive genetic algorithms.

The basic idea of the linear rank-based selection operator is as follows. Linear rank-based selection is to (1) line the solutions of the population into a sequence according to their fitness values, from large to small, and then (2) allocate a pre-designed sequence probability to each solution. The selection is based on this ranking rather than the absolute differences in fitness between different solutions. Without using the information of absolute differences of the solutions' fitness values, linear rank-based selection avoids changes of fitness scaling in the process of the population's evolution.

For a given population, assuming its scale is n , $P=\{a_1, a_2, \dots, a_n\}$, and the solutions' fitness values are given in a descending sequence $f(a_1) \geq f(a_2) \geq \dots \geq f(a_n)$, then each solution's selection probability is:

$$P(a_j) = \frac{1}{n} (\eta^+ - \frac{\eta^+ - \eta^-}{n-1} (j-1)) \quad j=1,2,\dots,n \quad (6)$$

where η^+ / n and η^- / n are respectively the probability of the best solutions and the worst solutions to be selected. As the population size is a constant, $\eta^- + \eta^+ = 2$, $\eta^- \geq 0$, and $1 \leq \eta^+ \leq 2$ must be satisfied. Because the rank-based selection probability is easy to control, and is especially suitable for dynamic adjusting, the selection pressure of the population can be changed timely in accordance with the effect of the population evolution. We can further improve the linear rank-based selection operator in the following way. The evolution of the population is divided into three stages: $[0, T_1]$, $[T_1, T_2]$, $[T_2, T]$, and apply different selection pressures as follows in these three stages:

$$\eta^+ = \begin{cases} \eta_1 & t \in [0, T_1] \\ \eta_2 & t \in (T_1, T_2] \\ \eta_3 & t \in (T_2, T] \end{cases} \quad (7)$$

where $T_1 = \alpha T$, $T_2 = (1 - \alpha)T$, $\alpha \in [0.35, 0.7]$, T is the largest generation of the evolution of the population, $\eta_1 \in [1.2, 1.6]$, $\eta_2 \in [1.4, 1.8]$, and $\eta_3 \in [1.8, 2.0]$.

Therefore, we can dynamically adjust the selection pressure in different stages of the evolution. In the early and middle stages of the evolution, we adopt smaller selection pressures and accept some solutions with low fitness, to maintain the population's diversity and facilitate global searching to the largest extent; and in the late stage, we adopt a larger selection pressure and accept no solutions with low fitness, in order to find the optimal solution from current solutions as soon as possible.

To make sure that the optimal solution is not to be destroyed, we adopt the elitist strategy^[12] at the same time, to make our adaptive genetic algorithm converge to the global optima in the end.

3.5 The TAGA Algorithm

By making use of each of the above-mentioned operators, the TAGA algorithm is given in pseudo-code as follows.

- Step 1.* Input parameters: size of the population, size of solutions, the maximal generation, the crossover point size, BNs, $\vec{\eta}$, $k_1 \sim k_4$;
- Step 2.* gen=0; //the initialization generation;
- Step 3.* chrom=initial_population(population_size, individual_size);
- Step 4.* objfv=obj_fun(bnet,chrom); //the value of the objective function
- Step 5.* fitnv=ranking(objfv); //calculate the fitness value
- Step 6.* Execute the adaptive linear rank-based selection;
- Step 7.* newchrom=adap_cross(newchrom,fitnv_mean1, fitnv_max1, fitnv1, k_2 , k_4);
- Step 8.* newchrom= adap_mut(newchrom, fitnv_mean1, fitnv_max, cro_point_size, k_1 , k_3);
- Step 9.* Recalculate the fitness value of the current population, and recombine it to the population to form a new population;
- Step 10.* gen=gen+1;
- Step 11.* If gen is smaller than the generation-maxgen, return to Step 5;
- Step 12.* Output the optimal node elimination sequence.

4 Experimental Results and Analysis

Definition 7.^[14] Under the circumstance of e, $X_e(s)$, as the on-line performance of strategy s, is defined as

$$X_e(s) = \frac{1}{T} \sum_{t=1}^T f_e(t) \quad (8)$$

In this formula, $f_e(t)$ is the average value of the objective function or the average fitness value at the time of t, under the e circumstance. From Definition 7, we know that the on-line performance of an algorithm shows the average value of the algorithm's performance since the very beginning when the algorithm starts to operate. It also reflects the dynamic performance of the algorithm.

Definition 8.^[14] Under the circumstance of e, $X_e(s)$, as the off-line performance of strategy s is defined as

$$X_e(s) = \frac{1}{T} \sum_{t=1}^T f_e^*(t) \quad (9)$$

In this formula, $f_e^*(t)$ is the best value of the objective function or the largest fitness value in the period [0,t] under the e circumstance. From Definition 8, we know that the off-line performance of an algorithm shows the cumulative mean of the best performance of each evolving generation in the process of the algorithm's operation. It reflects the convergence performance of the algorithm.

Three standard Bayesian networks are selected from [15] for experiments in this section, which are Asia, Car_diagnosis, Alarm. A simple genetic algorithm (SGA,

without adaptation on crossover and mutation probabilities), an adaptive genetic algorithm (AGA, by [10]), and TAGA are adopted in the triangulation of these BNs. In the end, we also compare them with the simulated annealing (SA) of K          [7].

Asia is a thorax diagnosis system with 8 nodes. Alarm is a relatively large medical diagnosis system that includes 37 nodes. Car_diagnosis is a vehicle failure diagnosis system which includes 18 nodes.

The following are the experimental results on the Car_diagnosis Bayesian network. The parameters of the three methods are as follows.

The size of the population is 46. maxgen=200. OX crossover and displacing mutation are adopted. In SGA, $P_c=0.8$, $P_m=0.006$; in TAGA, $\alpha=0.368$, $\eta_1=1.4$, $\eta_2=1.8$, $\eta_3=2.0$. The other parameters selected in AGA and TAGA are given in Table 1.

Table 1. Parameter settings

Algorithm	k_1	k_2	k_3	k_4
AGA	0.9	0.2	0.9	0.01
TAGA	1	0.2	1	0.1

According to Fig.1, TAGA maintains a better diversity of the population from 1 to 160 generations, which keeps certain changes of the best value of the fitness. It explains that TAGA searches for other promising basins. In contrast, the best values of the fitness for AGA and SGA change little from 80 and 110 respectively, which shows that the individuals of the population almost tend to be the same.

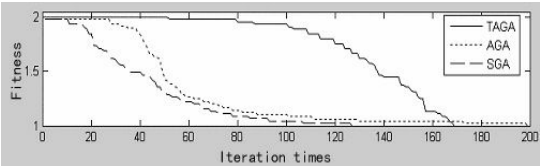


Fig. 1. Population's fitness

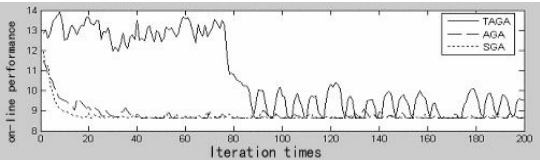


Fig. 2. On-line performance comparison

Fig.2 indicates that the changes of the average value of the objective function for AGA and SGA limit in a small range from 80 generations, which explains that the evolution of the population tends to stop. On the contrary, TAGA shows a better capability of global search. Fig.3 shows the off-line performance of these algorithms. Therefore, we have effectively avoided the premature convergence issue. In this algorithm, the ability to jump out of local optima is superior to the other two algorithms.

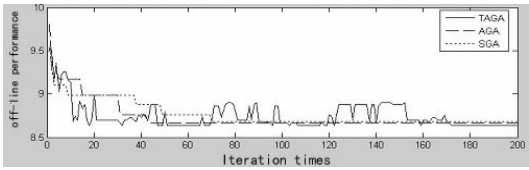


Fig. 3. Off-line performance comparison

In Figure 4, there is an average performance curve of the performance of the best individual value and the average value of the population with the TAGA algorithm, which was calculated over 5 times of experiments. From this figure, we can see the convergence speed of TAGA is comparatively slow in the early stage when performing global searching. In the late stage, convergence is achieved gradually to the present optima.

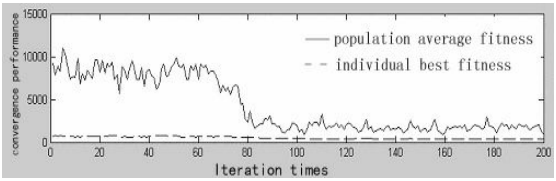


Fig. 4. Average convergence of TAGA

Now let us compare the results of the three genetic methods with the simulated annealing (SA) of Kjørulff. The figures in Table 2 show that the more complex the given network, the better the results of the TAGA algorithm, compared to the other algorithms.

Table 2. Minimal weight of junction tree of four algorithms

Algorithm	Asia	Car_diagnosis	Alarm
SA	40	416	1061
SGA	40	410	1032
AGA	40	406	1014
TAGA	40	398	996

5 Conclusion

This paper has proposed the TAGA algorithm to obtain a triangulation elimination sequence of Bayesian networks. Our experimental results have shown that TAGA can not only improve the on-line and off-line performances, but also outperform its rival algorithms. However, the global optima we have achieved are still on the sacrifice of a certain time performance. How to use parallel genetic algorithms to reduce the time complexity is a future direction for our current work, in order to gain a good balance between the optimal results and a good time performance.

References

1. J. Pearl, *Probabilistic Reasoning in Intelligence Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
2. P. Dagum and M. Luby, Approximating probabilistic inference using Bayesian networks is NP-hard, *Artificial Intelligence*. 60(1), 1993: 141-153
3. F. R. Bach and Michael I. Jordan, Thin Junction Trees, *NIPS 2001*: 569-576
4. D.J.Rose, R.E.Tarjan, and G.S.Lueker, Algorithmic aspects of vertex elimination on graphs, *SIAM J.Comput.* 5(2), 1976: 266-283
5. R.E. Tarjan and M. Yannakakis, Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, *SIAM J. Comput.* 13, 1984: 566-579
6. U. Kjærulff, Triangulation of graphs - Algorithms giving small total state space, *Technical Report R9009*, Department of Mathematics and Computer Science, University of Aalborg, Denmark, 1990
7. U. Kjærulff, Optimal decomposition of probabilistic networks by simulated annealing, *Stat. and Compu.* 2, 1992: 7-17
8. Pedro Larranaga, Cindy M.H. Kuijpers, Mikel Poza, and Roberto Murga, Decomposing Bayesian networks: triangulation of the moral graph with genetic algorithms, *Statistics and Computing* 1997 19-34
9. Thomas Bäck, Jeannette M. de Graaf, Joost N. Kok, and Walter A. Kisters, Theory of Genetic Algorithms. *Current Trends in Theoretical Computer Science* 2001: 546-578
10. Henri Luchian and Ovidiu Gheorghies: Integrated-Adaptive Genetic Algorithms. *ECAL* 2003: 635-642
11. Sangameswar Venkatraman and Gary G. Yen, A Generic Framework for Constrained Optimization Using Genetic Algorithms, *IEEE Transactions on Evolutionary Computation*, Vol.9, No.4, August 2005.
12. T. Blickle and L. Thiele, A Comparison of Selection Schemes used in Genetic Algorithms (2nd Edition), *Technical Report*, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, 1995.
13. W.X. Wen, Optimal decomposition of belief networks. In *Uncertainty in Artificial Intelligence* 1991, pp. 209-24. North-Holland, Amsterdam.
14. Ying Zhang and Yanqiu Liu, *Soft Computing Methodologies*, Beijing: Science Press, 2002.
15. Norsys Software Corp., <http://www.norsys.com>, 2006.

Intelligent Agents That Make Informed Decisions

John Debenham and Elaine Lawrence

University of Technology, Sydney, Australia
{`debenham`, `elaine`}@it.uts.edu.au

Abstract. Electronic markets with access to the Internet and the World Wide Web, are information-rich and require agents that can assimilate and use real-time information flows wisely. A new breed of “information-based” agents aims to meet this requirement. They are founded on concepts from information theory, and are designed to operate with information flows of varying and questionable integrity. These agents are part of a larger project that aims to make informed automated trading in applications such as eProcurement a reality.

1 Introduction

Despite the substantial advances in multiagent systems [1] and automated negotiation [2] [3], it is perhaps surprising that negotiation in electronic business [4] remains a substantially manual procedure. Electronic trading environments are awash with information, including information drawn from general resources such as the World Wide Web using smart retrieval technology [5]. Powerful agent architectures that are capable of flexible, autonomous action are well known [1]. We propose that rather than strive to make strategic, economically rational decisions, intelligent agents in electronic markets should capitalise on the real-time information flows and should aim to make ‘informed decisions’ that take account of the integrity of all relevant information. Traditional agent architectures, such as BDI, do not address directly the management of dynamic information flows of questionable integrity. This paper describes an agent architecture that has been designed specifically to operate in tandem with information discovery systems. This is part of our e-Market Framework that is available on the World Wide Web¹. This framework aims to make informed automated trading a reality, and aims to address the realities of electronic business — “its what you know that matters”. This work does not address all of the issues in automated trading [4]. For example, the work relies on developments in: XML and semantic web, secure data exchange, value chain management and financial services. Further the design of electronic marketplaces is not described here.

Intelligent agents that are built on an architecture designed specifically to handle real-time information flows are described in Sec. 2. The way in which these agents manage dynamic information flows is described in Sec. 3. The interaction

¹ <http://e-markets.org.au>

of more than one of these agents engaging in competitive negotiation is described in Sec. 4. Sec. 5 concludes.

2 Information-Based Agents

We have designed a new agent architecture founded on information theory. These “information-based” agents operate in real-time in response to market information flows. The central issues of trust in the execution of contracts is discussed in [6]. The “information-based” agent’s reasoning is based on a first-order logic world model that manages multi-issue negotiation as easily as single-issue.

2.1 Agent Architecture

An agent observes events in its environment and represents some of those observations in its world model as beliefs. As time passes, an agent may not be prepared to accept such beliefs as being “true”, and qualifies those representations with epistemic probabilities. Those qualified representations of prior observations are the agent’s *information*. Given this information, an agent may then choose to adopt goals and strategies. Those strategies may be based on game theory, for example. To enable the agent’s strategies to make good use of its information, tools from information theory are applied to summarise and process that information. Such an agent is called *information-based*.

An agent called Π is the subject of this discussion. Π engages in multi-issue negotiation with a set of other agents: $\{\Omega_1, \dots, \Omega_o\}$. The foundation for Π ’s operation is the information that is generated both by and because of its negotiation exchanges. Any message from one agent to another reveals information about the sender. Π uses ideas from information theory to process and summarise its information. If Π *does* know its utility function and if it aims to optimise its utility *then* Π may apply the principles of game theory to achieve its aim.

The information-based approach does not to reject utility optimisation — the selection of a strategy is secondary to the processing and summarising large amounts of uncertain information. In addition to the information derived from its opponents, Π has access to a set of information sources $\{\Theta_1, \dots, \Theta_t\}$ that may include the marketplace in which trading takes place, and general information sources such as news-feeds accessed via the Internet. Together, Π , $\{\Omega_1, \dots, \Omega_o\}$ and $\{\Theta_1, \dots, \Theta_t\}$ make up a multiagent system.

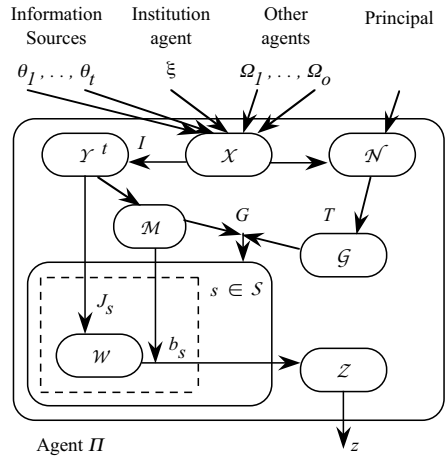


Fig. 1. Architecture of agent Π

Π has two languages: $\mathcal{C}$ and $\mathcal{L}$. $\mathcal{C}$ is an illocutionary-based language for communication. $\mathcal{L}$ is a first-order language for internal representation — precisely it is a first-order language with sentence probabilities optionally attached to each sentence representing Π 's epistemic belief in the truth of that sentence. Fig. 1 shows a high-level view of how Π operates. Messages expressed in $\mathcal{C}$ from $\{\Theta_i\}$ and $\{\Omega_i\}$ are received, time-stamped, source-stamped and placed in an *in-box* $\mathcal{X}$. The messages in $\mathcal{X}$ are then translated using an *import function* I into sentences expressed in $\mathcal{L}$ that have integrity decay functions (usually of time) attached to each sentence, they are stored in a *repository* $\mathcal{Y}^t$. And that is all that happens until Π triggers a goal.

Π triggers a goal, $g \in \mathcal{G}$, in two ways: first in response to a message received from an opponent $\{\Omega_i\}$ “I offer you €100 in exchange for a pallet of paper”, and second in response to some need, $\nu \in \mathcal{N}$, “we need to order some more paper”. In either case, Π is motivated by a need — either a need to strike a deal, or a general need to trade. Π 's goals could be short-term such as obtaining some information “what is the euro / dollar exchange rate?”, medium-term such as striking a deal with one of its opponents, or, rather longer-term such as building a (business) relationship with one of its opponents. So Π has a trigger mechanism T where: $T : \{\mathcal{X} \cup \mathcal{N}\} \rightarrow \mathcal{G}$.

For each goal that Π commits to, it has a mechanism, G , for selecting a strategy to achieve it where $G : \mathcal{G} \times \mathcal{M} \rightarrow \mathcal{S}$ where $\mathcal{S}$ is the strategy library. A *strategy* s maps an information base into an action, $s(\mathcal{Y}^t) = z \in \mathcal{Z}$. Given a goal, g , and the current state of the social model m^t , a strategy: $s = G(g, m^t)$. Each strategy, s , consists of a *plan*, b_s and a *world model* (construction and revision) *function*, J_s , that constructs, and maintains the currency of, the strategy's *world model* W_s^t that consists of a set of probability distributions. A *plan* derives the agent's next action, z , on the basis of the agent's world model for that strategy and the current state of the social model: $z = b_s(W_s^t, m^t)$, and $z = s(\mathcal{Y}^t)$. J_s employs two forms of entropy-based inference:

Maximum entropy inference. J_s^+ , first constructs an *information base* $\mathcal{I}_s^t$ as a set of sentences expressed in $\mathcal{L}$ derived from $\mathcal{Y}^t$, and then from $\mathcal{I}_s^t$ constructs the world model, W_s^t , as a set of complete probability distributions [using Eqn. 2 in Sec. 2.2 below].

Maximum relative entropy inference. Given a prior world model, W_s^u , where $u < t$, minimum relative entropy inference, J_s^- , first constructs the incremental information base $\mathcal{I}_s^{(u,t)}$ of sentences derived from those in $\mathcal{Y}^t$ that were received between time u and time t , and then from W_s^u and $\mathcal{I}_s^{(u,t)}$ constructs a new world model, W_s^t [using Eqn. 3 in Sec. 2.2 below].

2.2 Π 's Reasoning

Once Π has selected a plan $a \in \mathcal{A}$ it uses maximum entropy inference to derive the $\{D_i^s\}_{i=1}^n$ [see Fig. 1] and minimum relative entropy inference to update those distributions as new data becomes available. *Entropy*, $\mathbb{H}$, is a measure of uncertainty [7] in a probability distribution for a discrete random variable X :

$\mathbb{H}(X) \triangleq -\sum_i p(x_i) \log p(x_i)$ where $p(x_i) = \mathbb{P}(X = x_i)$. Maximum entropy inference is used to derive sentence probabilities for that which is not known by constructing the “maximally noncommittal” [8] probability distribution, and is chosen for its ability to generate complete distributions from sparse data.

Let $\mathcal{G}$ be the set of all positive ground literals that can be constructed using Π ’s language $\mathcal{L}$. A *possible world*, v , is a valuation function: $\mathcal{G} \rightarrow \{\top, \perp\}$. $\mathcal{V}|\mathcal{K}^s = \{v_i\}$ is the set of all possible worlds that are consistent with Π ’s knowledge base $\mathcal{K}^s$ that contains statements which Π believes are true. A *random world* for $\mathcal{K}^s$, $W|\mathcal{K}^s = \{p_i\}$ is a probability distribution over $\mathcal{V}|\mathcal{K}^s = \{v_i\}$, where p_i expresses Π ’s degree of belief that each of the possible worlds, v_i , is the actual world. The *derived sentence probability* of any $\sigma \in \mathcal{L}$, with respect to a random world $W|\mathcal{K}^s$ is:

$$(\forall \sigma \in \mathcal{L}) \mathbb{P}_{\{W|\mathcal{K}^s\}}(\sigma) \triangleq \sum_n \{p_n : \sigma \text{ is } \top \text{ in } v_n\} \quad (1)$$

The agent’s *belief set* $\mathcal{B}_t^s = \{\Omega_j\}_{j=1}^M$ contains statements to which Π attaches a *given sentence probability* $\mathbb{B}(\cdot)$. A random world $W|\mathcal{K}^s$ is *consistent* with $\mathcal{B}_t^s$ if: $(\forall \Omega \in \mathcal{B}_t^s)(\mathbb{B}(\Omega) = \mathbb{P}_{\{W|\mathcal{K}^s\}}(\Omega))$. Let $\{p_i\} = \{\overline{W}|\mathcal{K}^s, \mathcal{B}_t^s\}$ be the “maximum entropy probability distribution over $\mathcal{V}|\mathcal{K}^s$ that is consistent with $\mathcal{B}_t^s$ ”. Given an agent with $\mathcal{K}^s$ and $\mathcal{B}_t^s$, *maximum entropy inference* states that the *derived sentence probability* for any sentence, $\sigma \in \mathcal{L}$, is:

$$(\forall \sigma \in \mathcal{L}) \mathbb{P}_{\{\overline{W}|\mathcal{K}^s, \mathcal{B}_t^s\}}(\sigma) \triangleq \sum_n \{p_n : \sigma \text{ is } \top \text{ in } v_n\} \quad (2)$$

From Eqn. 2, each belief imposes a linear constraint on the $\{p_i\}$. The maximum entropy distribution: $\arg \max_{\underline{p}} \mathbb{H}(\underline{p})$, $\underline{p} = (p_1, \dots, p_N)$, subject to $M + 1$ linear constraints:

$$g_j(\underline{p}) = \sum_{i=1}^N c_{ji} p_i - \mathbb{B}(\Omega_j) = 0, \quad j = 1, \dots, M. \quad g_0(\underline{p}) = \sum_{i=1}^N p_i - 1 = 0$$

where $c_{ji} = 1$ if Ω_j is $\top$ in v_i and 0 otherwise, and $p_i \geq 0, i = 1, \dots, N$, is found by introducing Lagrange multipliers, and then obtaining a numerical solution using the multivariate Newton-Raphson method. In the subsequent subsections we’ll see how an agent updates the sentence probabilities depending on the type of information used in the update.

Given a prior probability distribution $\underline{q} = (q_i)_{i=1}^n$ and a set of constraints C , the *principle of minimum relative entropy* chooses the posterior probability distribution $\underline{p} = (p_i)_{i=1}^n$ that has the least *relative entropy*² with respect to $\underline{q}$:

$$\{\underline{W}|\underline{q}, C\} \triangleq \arg \min_{\underline{p}} \sum_{i=1}^n p_i \log \frac{p_i}{q_i}$$

² Otherwise called *cross entropy* or the *Kullback-Leibler* distance between the two probability distributions.

and that satisfies the constraints. This may be found by introducing Lagrange multipliers as above. Given a prior distribution $\underline{q}$ over $\{v_i\}$ — the set of all possible worlds, and a set of constraints C (that could have been derived as above from a set of new beliefs) *minimum relative entropy inference* states that the derived sentence probability for any sentence, $\sigma \in \mathcal{L}$, is:

$$(\forall \sigma \in \mathcal{L}) \mathbb{P}_{\{\underline{W}|\underline{q}, C\}}(\sigma) \triangleq \sum_n \{ p_n : \sigma \text{ is } \top \text{ in } v_n \} \quad (3)$$

where $\{p_i\} = \{\underline{W}|\underline{q}, C\}$. The principle of minimum relative entropy is a generalisation of the principle of maximum entropy. If the prior distribution $\underline{q}$ is uniform, then the relative entropy of $\underline{p}$ with respect to $\underline{q}$, $\underline{p}||\underline{q}$, differs from $-\mathbb{H}(\underline{p})$ only by a constant. So the principle of maximum entropy is equivalent to the principle of minimum relative entropy with a uniform prior distribution.

3 Managing Dynamic Information Flows

The illocutions in the communication language $\mathcal{C}$ include information, $[info]$. The information received from general information sources will be expressed in terms defined by Π 's ontology. We define an *ontology signature* as a tuple $S = (C, R, \leq, \sigma)$ where C is a finite set of concept symbols (including basic data types); R is a finite set of relation symbols; $\leq$ is a reflexive, transitive and anti-symmetric relation on C (a partial order); and, $\sigma : R \rightarrow C^+$ is the function assigning to each relation symbol its arity. Concepts play the role of *type*, and the is-a hierarchy is the notion of subtype. Thus, type inference mechanisms can be used to type all symbols appearing in expressions. We assume that Π makes at least part of that ontology public so that the other agents $\{\Omega_1, \dots, \Omega_o\}$ may communicate $[info]$ that Π can understand. Ω 's *reliability* is an estimate of the extent to which this $[info]$ is correct.

The only restriction on incoming $[info]$ is that it is expressed in terms of the ontology — this is very general. However, the way in which $[info]$ is used is completely specific — it will be represented as a set of linear constraints on one or more probability distributions in the world model. A chunk of $[info]$ may not be directly related to one of Π 's chosen distributions or may not be expressed naturally as constraints, and so some inference machinery is required to derive these constraints — this inference is performed by model building functions, J_s , that have been activated by a plan s chosen by Π . $J_s^D([info])$ denotes the set of constraints on distribution D derived by J_s from $[info]$.

3.1 Updating the World Model with $[info]$

The procedure for updating the world model as $[info]$ is received follows. If at time u , Π receives a message containing $[info]$ it is time-stamped and source-stamped $[info]_{(\Omega, \Pi, u)}$, and placed in a repository $\mathcal{Y}^t$. If Π has an active plan, s , with model building function, J_s , then J_s is applied to $[info]_{(\Omega, \Pi, u)}$ to derive constraints on some, or none, of Π 's distributions. The extent to which those

constraints are permitted to effect the distributions is determined by a value for the *reliability* of Ω , $R^t(\Pi, \Omega, O([info]))$, where $O([info])$ is the ontological context of $[info]$.

An agent may have models of integrity decay for some particular distributions, but general models of integrity decay for, say, a chunk of information taken at random from the World Wide Web are generally unknown. However the values to which decaying integrity should tend in time *are* often known. For example, a prior value for the truth of the proposition that a “22 year-old male will default on credit card repayment” is well known to banks. If Π attaches such prior values to a distribution D they are called the *decay limit distribution* for D , $(d_i^D)_{i=1}^n$. No matter how integrity of $[info]$ decays, in the absence of any other relevant information it should decay to the decay limit distribution. If a distribution with n values has no decay limit distribution then integrity decays to the maximum entropy value $\frac{1}{n}$. In other words, the maximum entropy distribution is the default decay limit distribution.

In the absence of new $[info]$ the integrity of distributions decays. If $D = (q_i)_{i=1}^n$ then we use a geometric model of decay:

$$q_i^{t+1} = (1 - \rho^D) \times d_i^D + \rho^D \times q_i^t, \text{ for } i = 1, \dots, n \quad (4)$$

where $\rho^D \in (0, 1)$ is the decay rate. This raises the question of how to determine ρ^D . Just as an agent may know the decay limit distribution it may also know something about ρ^D . In the case of an information-overfed agent there is no harm in conservatively setting ρ^D “a bit on the low side” as the continually arriving $[info]$ will sustain the estimate for D .

We now describe how new $[info]$ is imported to the distributions. A single chunk of $[info]$ may effect a number of distributions. Suppose that a chunk of $[info]$ is received from Ω and that Π attaches the epistemic belief probability $R^t(\Pi, \Omega, O([info]))$ to it. Each distribution models a facet of the world. Given a distribution $D^t = (q_i^t)_{i=1}^n$, q_i^t is the probability that the possible world ω_i for D is the true world for D . The effect that a chunk $[info]$ has on distribution D is to enforce the set of linear constraints on D , $J_s^D([info])$. If the constraints $J_s^D([info])$ are taken by Π as valid then Π could update D to the posterior distribution $(p_i^{[info]})_{i=1}^n$ that is the distribution with least relative entropy with respect to $(q_i^t)_{i=1}^n$ satisfying the constraint:

$$\sum_i \{p_i^{[info]} : J_s^D([info]) \text{ are all } \top \text{ in } \omega_i\} = 1. \quad (5)$$

But $R^t(\Pi, \Omega, O([info])) = r \in [0, 1]$ and Π should only treat the $J_s^D([info])$ as valid if $r = 1$. In general r determines the extent to which the effect of $[info]$ on D is closer to $(p_i^{[info]})_{i=1}^n$ or to the prior $(q_i^t)_{i=1}^n$ distribution by:

$$p_i^t = r \times p_i^{[info]} + (1 - r) \times q_i^t \quad (6)$$

But, we should only permit a new chunk of $[info]$ to influence D if doing so gives us new information. For example, if 5 minutes ago a trusted agent advises Π

that the interest rate will go up by 1%, and 1 minute ago a very unreliable agent advises Π that the interest rate may go up by 0.5%, then the second unreliable chunk should not be permitted to ‘overwrite’ the first. We capture this by only permitting a new chunk of $[info]$ to be imported if the resulting distribution has more information *relative* to the decay limit distribution than the existing distribution has. Precisely, this is measured using the Kullback-Leibler distance measure — this is just one criterion for determining whether the $[info]$ should be used — and $[info]$ is only used if:

$$\sum_{i=1}^n p_i^t \log \frac{p_i^t}{d_i^D} > \sum_{i=1}^n q_i^t \log \frac{q_i^t}{d_i^D} \quad (7)$$

In addition, we have described in Eqn. 4 how the integrity of each distribution D will decay in time. Combining these two into one result, distribution D is revised to:

$$q_i^{t+1} = \begin{cases} (1 - \rho^D) \times d_i^D + \rho^D \times p_i^t & \text{if usable } [info] \text{ is received at time } t \\ (1 - \rho^D) \times d_i^D + \rho^D \times q_i^t & \text{otherwise} \end{cases}$$

for $i = 1, \dots, n$, and decay rate ρ^D as before.

3.2 Information Reliability

Sec. 3.1 relies on an estimate of $R^t(\Pi, \Omega, O([info]))$. This estimate is constructed by measuring the ‘error’ in observed information as the error in the effect that information has on each of Π ’s distributions. Suppose that a chunk of $[info]$ is received from agent Ω at time s and is verified at some later time t . For example, a chunk of information could be “the interest rate will rise by 0.5% next week”, and suppose that the interest rate actually rises by 0.25% — call that correct information $[fact]$. What does all this tell agent Π about agent Ω ’s reliability? Consider one of Π ’s distributions D that is $\{q_i^s\}$ at time s . Let $(p_i^{[info]})_{i=1}^n$ be the minimum relative entropy distribution given that $[info]$ has been received as calculated in Eqn. 5, and let $(p_i^{[fact]})_{i=1}^n$ be that distribution if $[fact]$ had been received instead. Suppose that the reliability estimate for distribution D was R_D^s . This section is concerned with what R_D^s should have been in the light of knowing *now*, at time t , that $[info]$ should have been $[fact]$, and how that knowledge effects our current reliability estimate for D , $R^t(\Pi, \Omega, O([info]))$.

The idea of Eqn. 6, is that the current value of r should be such that, *on average*, $(p_i^s)_{i=1}^n$ will be seen to be “close to” $(p_i^{[fact]})_{i=1}^n$ when we eventually discover $[fact]$ — no matter whether or not $[info]$ was used to update D , as determined by the acceptability test in Eqn. 7 at time s . That is, given $[info]$, $[fact]$ and the prior $(q_i^s)_{i=1}^n$, calculate $(p_i^{[info]})_{i=1}^n$ and $(p_i^{[fact]})_{i=1}^n$ using Eqn. 5. Then the *observed reliability* for distribution D , $R_D^{([info]||[fact])}$, on the basis of the verification of $[info]$ with $[fact]$ is the value of r that minimises the Kullback-Leibler distance between $(p_i^s)_{i=1}^n$ and $(p_i^{[fact]})_{i=1}^n$:

$$\arg \min_r \sum_{i=1}^n (r \cdot p_i^{[info]} + (1-r) \cdot q_i^s) \log \frac{r \cdot p_i^{[info]} + (1-r) \cdot q_i^s}{p_i^{[fact]}}$$

If $E^{[info]}$ is the set of distributions that $[info]$ affect, then the overall *observed reliability* on the basis of the verification of $[info]$ with $[fact]$ is: $R^{([info]||[fact])} = 1 - (\max_{D \in E^{[info]}} |1 - R_D^{([info]||[fact])}|)$. Then for each ontological context o_j , at time t when, perhaps, a chunk of $[info]$, with $O([info]) = o_k$, may have been verified with $[fact]$:

$$R^{t+1}(\Pi, \Omega, o_j) = (1 - \rho) \times R^t(\Pi, \Omega, o_j) + \rho \times R^{([info]||[fact])} \times \text{Sem}(o_j, o_k)$$

where $\text{Sem}(\cdot, \cdot) : O \times O \rightarrow [0, 1]$ measures the semantic distance [9] between two sections of the ontology, and ρ is the learning rate. Over time, Π notes the ontological context of the various chunks of $[info]$ received from Ω and over the various ontological contexts calculates the relative frequency, $P^t(o_j)$, of these contexts, $o_j = O([info])$. This leads to an overall expectation of the *reliability* that agent Π has for agent Ω : $R^t(\Pi, \Omega) = \sum_j P^t(o_j) \times R^t(\Pi, \Omega, o_j)$.

4 Negotiation

For illustration Π 's communication language [10] is restricted to the illocutions: Offer($\cdot$), Accept($\cdot$), Reject($\cdot$) and Withdraw($\cdot$). The simple strategies that we will describe all use the same world model function, J_s , that maintains the following two probability distributions as their world model:

- $\mathbb{P}^t(\Pi \text{Acc}(\Pi, \Omega, \nu, \delta))$ — the strength of belief that Π has in the proposition that she should accept the proposal $\delta = (a, b)$ from agent Ω in satisfaction of need ν at time t , where a is Π 's commitment and b is Ω 's commitment. $\mathbb{P}^t(\Pi \text{Acc}(\Pi, \Omega, \nu, \delta))$ is estimated from:
 1. $\mathbb{P}^t(\text{Satisfy}(\Pi, \Omega, \nu, \delta))$ a subjective evaluation (the strength of belief that Π has in the proposition that the expected outcome of accepting the proposal will satisfy some of her needs).
 2. $\mathbb{P}^t(\text{Fair}(\delta))$ an objective evaluation (the strength of belief that Π has in the proposition that the proposal is a “fair deal” in the open market).
 3. $\mathbb{P}^t(\Pi \text{CanDo}(a))$ an estimate of whether Π will be able to meet her commitment a at contract execution time.

These three arrays of probabilities are estimated by importing relevant information, $[info]$, as described in Sec. 3.

- $\mathbb{P}^t(\Omega \text{Acc}(\beta, \alpha, \delta))$ — the strength of belief that Π has in the proposition that Ω would accept the proposal δ from agent Π at time t . Every time that Ω submits a proposal she is revealing information about what she is prepared to accept, and every time she rejects a proposal she is revealing information about what she is *not* prepared to accept. Eg: having received the stamped illocution $\text{Offer}(\Omega, \Pi, \delta)_{(\Omega, \Pi, u)}$, at time $t > u$, Π may believe that $\mathbb{P}^t(\Omega \text{Acc}(\Omega, \Pi, \delta)) = \kappa$ this is used as a constraint on $\mathbb{P}^{t+1}(\Omega \text{Acc}(\cdot))$ which is calculated using Eqn. 3.

Negotiation is an information revelation process. The agents described in Sec. 2 are primarily concerned with the integrity of their information, and then when they have acquired sufficient information to reduce their uncertainty to an acceptable level they are concerned with acting strategically within the surrounding uncertainty. A basic conundrum in any offer-exchange bargaining is: it is impossible to force your opponent to reveal information about their position without revealing information about your own position. Further, by revealing information about your own position you may change your opponents position — and so on.³ This infinite regress, of speculation and counter-speculation, is avoided here by ignoring the internals of the opponent and by focussing on what is known for certain — that is: *what* information is contained in the signals received and *when* did those signals arrive.

4.1 Negotiation Strategies

An agent's strategy s is a function of the information $\mathcal{Y}^t$ that it has at time t . Four simple strategies make offers only on the basis of $\mathbb{P}^t(\Pi\text{Acc}(\Pi, \Omega, \nu, \delta))$, Π 's acceptability threshold γ , and $\mathbb{P}^t(\Omega\text{Acc}(\Omega, \Pi, \delta))$. The greedy strategy s^+ chooses:

$$\arg \max_{\delta} \{ \mathbb{P}^t(\Pi\text{Acc}(\Pi, \Omega, \nu, \delta)) \mid \mathbb{P}^t(\Omega\text{Acc}(\Omega, \Pi, \delta)) \gg 0 \},$$

it is appropriate when Π believes Ω is desperate to trade.

The *expected-acceptability-to- Π -optimizing strategy* s^* chooses:

$$\arg \max_{\delta} \{ \mathbb{P}^t(\Omega\text{Acc}(\Omega, \Pi, \delta)) \times \mathbb{P}^t(\Pi\text{Acc}(\Pi, \Omega, \nu, \delta)) \mid \mathbb{P}^t(\Pi\text{Acc}(\Pi, \Omega, \nu, \delta)) \geq \gamma \}$$

when Π is confident and not desperate to trade. The strategy s^- chooses:

$$\arg \max_{\delta} \{ \mathbb{P}^t(\Omega\text{Acc}(\Omega, \Pi, \delta)) \mid \mathbb{P}^t(\Pi\text{Acc}(\Pi, \Omega, \nu, \delta)) \geq \gamma \}$$

it optimises the likelihood of trade — when Π is keen to trade without compromising its own standards of acceptability.

An approach to issue-tradeoffs is described in [11]. The bargaining strategy described there attempts to make an acceptable offer by “walking round” the iso-curve of Π 's previous offer δ' (that has, say, an acceptability of $\gamma_{\delta'} \geq \gamma$) towards Ω 's subsequent counter offer. In terms of the machinery described here, an analogue is to use the strategy s^- :

$$\arg \max_{\delta} \{ \mathbb{P}^t(\Omega\text{Acc}(\Omega, \Pi, \delta)) \mid \mathbb{P}^t(\Pi\text{Acc}(\Pi, \Omega, \nu, \delta)) \geq \gamma_{\delta'} \}$$

with $\gamma = \gamma_{\delta'}$. This is reasonable for an agent that is attempting to be accommodating without compromising its own interests. The complexity of the strategy in [11] is linear with the number of issues. The strategy described here does not have that property, but it benefits from using $\mathbb{P}^t(\Omega\text{Acc}(\Omega, \Pi, \delta))$ that contains foot prints of the prior offer sequence — estimated by repeated use of Eqn. 3 — in that distribution more recent data gives estimates with greater certainty.

³ This is a reminiscent of Werner Heisenberg's indeterminacy relation, or *unbestimmtheitsrelationen*: “you can't measure one feature of an object without changing another” — with apologies.

5 Conclusions

Electronic marketplaces with access to the Internet and the World Wide Web are awash with dynamic information flows including hard market data and soft textual data such as news feeds. Structured and unstructured data mining methods may be applied to identify, analyse, condense and deliver signals from this data whose integrity may be questionable [5]. The key to automating trading is to build intelligent agents that are ‘informed’, that can proactively acquire information to reduce uncertainty, that can estimate the integrity of real-time information flows, and can use uncertain information as a foundation for strategic decision-making [4]. An ‘information-based’ agent architecture has been described, that is founded on ideas from information theory, and has been developed specifically for this purpose. It is part of our eMarket platform¹ that also includes information discovery systems, a “virtual institution” system in a collaborative research project with “Institut d’Investigacio en Intel·ligencia Artificial⁴”, Spanish Scientific Research Council, UAB, Barcelona, Spain.

References

1. Wooldridge, M.: *Multiagent Systems*. Wiley (2002)
2. Kraus, S.: *Strategic Negotiation in Multiagent Environments*. MIT Press (2001)
3. Rosenschein, J.S., Zlotkin, G.: *Rules of Encounter*. The MIT Press, Cambridge, USA (1994)
4. Lawrence, E., Newton, S., Corbitt, B., Lawrence, J., Dann, S., Thanasankit, T.: *Internet Commerce — Digital Models for Business*. 3rd edn. John Wiley and Sons, Inc. (2003)
5. Zhang, D., Simoff, S.: Informing the Curious Negotiator: Automatic news extraction from the Internet. In Williams, G., Simoff, S., eds.: *Data Mining: Theory, Methodology, Techniques, and Applications*. Springer-Verlag: Heidelberg, Germany (2006) 176 – 191
6. Sierra, C., Debenham, J.: An information-based model for trust. In Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M., Wooldridge, M., eds.: *Proceedings Fourth International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2005, Utrecht, The Netherlands*, ACM Press, New York (2005) 497 – 504
7. MacKay, D.: *Information Theory, Inference and Learning Algorithms*. Cambridge University Press (2003)
8. Jaynes, E.: *Probability Theory — The Logic of Science*. Cambridge University Press (2003)
9. Li, Y., Bandar, Z.A., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering* **15** (2003) 871 – 882
10. Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Sierra, C., Wooldridge, M.: Automated negotiation: Prospects, methods and challenges. *International Journal of Group Decision and Negotiation* **10** (2001) 199–215
11. Faratin, P., Sierra, C., Jennings, N.: Using similarity criteria to make issue trade-offs in automated negotiation. *Journal of Artificial Intelligence* **142** (2003) 205–237

⁴ <http://www.iiia.csic.es/>

Using Intelligent Agents in e-Government for Supporting Decision Making About Service Proposals

Pasquale De Meo, Giovanni Quattrone, and Domenico Ursino

DIMET, Università Mediterranea di Reggio Calabria, Via Graziella,
Località Feo di Vito, 89060 Reggio Calabria, Italy
`demeo@unirc.it`, `quattrone@unirc.it`, `ursino@unirc.it`

Abstract. This paper aims at studying the possibility of exploiting the Intelligent Agent technology in e-government for supporting the decision making activity of government agencies. Specifically, it proposes a system to assist managers of a government agency, who plan to propose a new service, to identify those citizens that could gain the highest benefit from it. The paper illustrates the proposed system and reports some experimental results.

1 Introduction

The term “e-government” generally indicates the exploitation of Information and Communication Technologies for improving the interaction between citizens and government agencies [9]. For instance, e-government systems simplify the citizen’s access to government services, allow government agencies to effectively deliver their services, and enhance citizen participation in government and decision-making activities. In the last few years, both local and national governments worldwide are planning and realizing e-government programs. These initiatives have been generally welcomed by citizens, and several studies point out that many citizens and businesses require online services from their government [1,9,10].

These considerations motivate the enormous, both technological and scientific, efforts performed in the last few years for improving the range and the quality of the services offered on-line by government agencies.

One of the most interesting research directions consists of defining architectures and frameworks for supporting citizens in their access to e-government services; these systems should recommend to citizens those services appearing to be the most relevant and interesting for them, according to their needs and past behaviour [4]. Another interesting research effort is devoted to support the managers of government agencies to make their decisions about the definition and the provision of services to be offered to citizens [2,3].

This paper aims at providing a contribution in this setting; specifically, it proposes a system for supporting decision making activities of the managers of government agencies. The reasoning underlying our system is the following: in

the last years, government agencies have made a significant effort to plan and realize initiatives that can really improve the living conditions of citizens. These initiatives have generally a strong impact on public expenditure; as a consequence, it appears extremely relevant, in the politics of most (or all) countries, the capability of matching *social equity* and *service efficiency*.

Our system can supply an important support in this context. In fact, it aims at assisting the managers of a government agency, who plan to propose a new service, to identify those citizens that could gain the highest benefit from it.

The specific connotations of the e-government scenarios and the main features of our system make the exploitation of the Intelligent Agent technology extremely appropriate. Intelligent Agents have been extensively applied in the past for handling the distributed management of a wide variety of services (e.g., e-commerce, e-learning, e-recruitment, and so on). Their adoption in the context of e-government, instead, received less attention. The goal of this paper is to show, by presenting a system, that the Intelligent Agent technology not only can be applied, but also can provide important benefits, in the e-government context.

Our system associates an appropriate user profile [6] with each citizen accessing it; user profiles are suitable data structures storing demographical data, preferences, needs and past behaviours of involved users. In their construction and maintenance a key role is played by the learning capability of Intelligent Agents. In the past, user profiles have been extensively exploited in various application contexts, such as Web search, query answering and e-commerce; however, they received quite a limited attention in the e-government research field, especially for realizing decision support systems helping the managers of Public Administration offices.

Actually, in our opinion, user profiles can improve the effectiveness and the accuracy of an e-government system. Specifically, their exploitation enables to distinguish the behaviour and the needs of different users and to adapt the proposals of an e-government system to them. This can be extremely important for both citizens (who might receive proposals only relative to services they really need) and government agencies (that might find the users having the maximum benefit from a new service and/or might design new services best matching user expectations). As it will be clear in the following, in our specific context, the information stored in user profiles allows our system to compute how much a citizen might benefit from the activation of a service and how much a government office should pay for providing a citizen with the service itself.

As a further, interesting feature, user profiles in the context of e-government are particularly rich and detailed since many interesting data about citizens are already owned by Public Administration offices. In our opinion this fact is particularly important since the richness of user profiles is a key factor in software systems based on User Modelling.

The outline of the paper is as follows. Section 2 presents a description of the proposed system. Some experiments performed to test its accuracy are illustrated in Section 3. Finally, in Section 4, we draw our conclusions.

2 Description of the Proposed System

The general architecture of our system is shown in Figure 1. From this figure it is possible to observe that our system is characterized by three typologies of agents, namely: (i) User Agents (hereafter, *UA*), that support users in their access to e-government services; (ii) Service Manager Agents (hereafter, *SMA*), that detect, among all available services, those appearing to be the closest to user needs and profiles; (iii) Public Administration Agents (hereafter, *PAA*), that support the managers of Public Administration offices in their decision making activities. Our system exploits also a Service Database *SD*, that stores and manages information about the various services provided by the Public Administration offices, and a User Profile Database *UPD*, that stores and handles information about user profiles.

In the following sub-sections we describe in detail all these components.

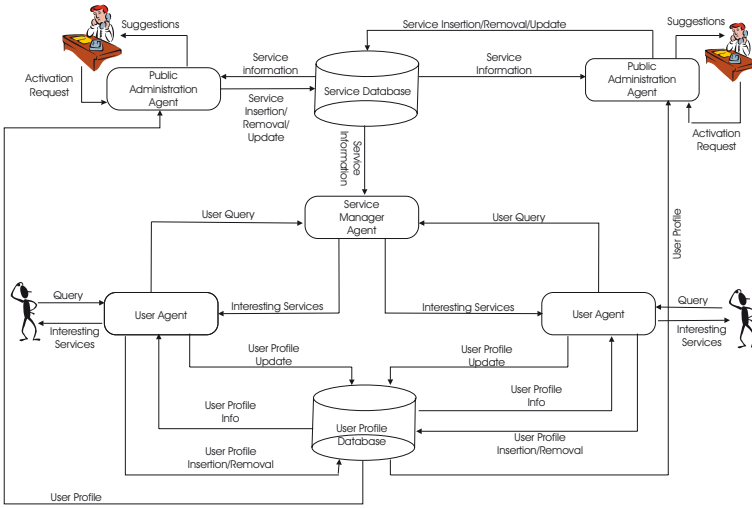


Fig. 1. Architecture of the proposed system

2.1 Support Databases

The User Profile Database *UPD* stores the profiles of the citizens accessing our system. A User Profile UP_i consists of a pair $\langle DS_i, KS_i \rangle$, where: (i) DS_i stores the *demographic data* of U_i , e.g., his Social Security Number, his age, his education level, his yearly income, and so on; (ii) KS_i is a *set of keywords* representing the past interests of U_i ; a coefficient in the real interval $[0, 1]$ is associated with each keyword, specifying the interest degree of U_i for it; this coefficient is computed by applying the reasoning illustrated in [7].

The Service Database *SD* stores information about the services offered by government agencies. A service S_k is represented by means of a pair $\langle Id_k, FS_k \rangle$,

where: (i) Id_k is a code identifying it; (ii) FS_k is a set of features associated with it, i.e., a set of keywords describing its main peculiarities.

2.2 The User Agent

A User Agent UA_i is associated with a user (i.e., a citizen) U_i . Its support data structure stores the profile UP_i of U_i .

UA_i is activated by U_i when he wants to know if there exist new services interesting for him and, in the affirmative case, to access them. First UA_i retrieves UP_i from UPD and stores it in its support data structure. Then, it allows U_i to pose a query Q_i , consisting of a set of keywords specifying the services he presently desires. After this, it sends the pair $\langle Q_i, UP_i \rangle$ to SMA , which is in charge of processing Q_i and constructing a list $\mathcal{SL}_i$ of services possibly satisfying U_i . SMA constructs $\mathcal{SL}_i$ and sends it to UA_i that proposes it to U_i ; the user can select the most relevant services, according to his needs.

On the basis of the choices of U_i , UA_i suitably updates UP_i . Such an activity consists of two steps, namely:

- *Insertion/Update of interests.* In this case UA_i examines both Q_i and the features of the services of $\mathcal{SL}_i$ chosen by U_i . It adds into KS_i each keyword/feature not already present therein, and associates a suitable, initial interest degree with it. In addition, for each keyword/feature already present into KS_i , it suitably updates the corresponding interest degree (some possible policies for performing this last task are described in [7]).
- *Pruning of old interests.* In this case UA_i removes from KS_i all keywords appearing non longer interesting for U_i on the basis of their interest degree. The pruning activity allows UA_i to maintain UP_i up-to-date and, at the same time, avoids the excessive growth of the size of UP_i .

Finally, when U_i ends his activities, UA_i inserts the updated UP_i in UPD .

Due to space limitations, we do not provide in this paper further details about the policies exploited by our system for updating user profiles; however, the interested reader can refer to [8] for a general overview of them.

2.3 The Service Manager Agent

The Service Manager Agent SMA is in charge of answering the queries submitted by users. Its support data structure consists of a list of services; the description of each service is analogous to that illustrated in Section 2.1.

SMA is activated by a User Agent UA_i each time a user U_i wants to access available services; it receives a pair $\langle Q_i, UP_i \rangle$, where Q_i is a query specifying the user desires and UP_i represents his profile.

First it constructs the list $\mathcal{SL}_i$ of services best matching Q_i and UP_i ; to this purpose it initially applies classical Information Retrieval techniques to select from SD the list $\mathcal{SL}_i^{temp}$ of services matching Q_i and being compatible with the demographic data of U_i . Then, for each service $S_k \in \mathcal{SL}_i^{temp}$, it computes a score stating its relevance for U_i ; the score computation takes into account the

features of S_k and the keywords of UP_i . After this, it constructs $\mathcal{SL}_i$ by selecting from $\mathcal{SL}_i^{temp}$ those services having the highest scores.

After $\mathcal{SL}_i$ has been constructed, SMA sends it to UA_i .

2.4 The Public Administration Agent

The Public Administration Agent PAA supports the manager of a government agency in his decision making activity, specifically in identifying users that can gain the maximum benefit from the activation of a service S_k .

PAA uses a suitable support data structure that stores: (i) a pair $\langle Id_k, FS_k \rangle$, analogous to that described in Section 2.1, concerning information about a new service S_k that must be activated; (ii) the maximum budget B_k available for the activation of S_k (*activation budget*).

First, for each user U_i , PAA computes a real coefficient $b_{ik} \in [0, 1]$, denoting the benefit that U_i would gain from the activation of S_k , and a real coefficient c_{ik} , representing the cost that a government agency should sustain to provide U_i with S_k .

The computation of b_{ik} is performed on the basis of the following reasoning:

- The more the set of features FS_k of S_k is similar to the set of keywords KS_i of UP_i , the more S_k can be considered helpful for U_i . In order to quantify the similarity between FS_k and KS_i we exploit the *Jaccard coefficient*: $\phi(UP_i, S_k) = \frac{|KS_i \cap FS_k|}{|KS_i \cup FS_k|}$. Observe that the values of ϕ belong to the real interval $[0, 1]$.
- The benefit of S_k for U_i depends on his economical and demographical conditions. In fact, in real life, the same service might have a different impact on citizens having different incomes. In order to formalize such an observation, it is necessary to construct a function ψ that receives information about the profile of U_i and the service S_k and returns a coefficient, in the real interval $[0, 1]$, specifying the benefit of S_k for U_i .

The definition of this function strictly depends on the welfare principles adopted by a Public Administration and, consequently, it might be different in the various countries and/or regions. A possible, quite general, formula is $\psi(UP_i, S_k) = K_{ik}^I \times K_{ik}^D \times K_{ik}^C$, where: (i) K_{ik}^I is a coefficient, in the real interval $[0, 1]$, depending on both the yearly income of U_i and the impact that the presence/absence of S_k might have on him; (ii) K_{ik}^D is a coefficient, in the real interval $[0, 1]$, taking into account the disabilities/diseases of U_i and the relevance of S_k for facing them; (iii) K_{ik}^C is a coefficient, in the real interval $[0, 1]$, considering the number of children of U_i and the relevance of S_k in this context.

In our tests, for determining the tables stating the values of these coefficients, we have considered the Italian Central Government welfare policies and we have required the support of a domain expert. Some of these tables can be found at the address <http://www.ing.unirc.it/ursino/ismis06/tables.html>. Clearly, a more sophisticated definition of ψ , taking into account other parameters about U_i , as well as the role of S_k on each of these parameters, could

be considered. In addition, both the previous definition and other, more sophisticated, ones could be adapted to other, possibly not Italian, contexts.

On the basis of the previous observations the benefit b_{ik} directly depends on ϕ and ψ , i.e., $b_{ik}(UP_i, S_k) = \phi(UP_i, S_k) \times \psi(UP_i, S_k)$.

Let us now consider c_{ik} , i.e., the cost a government agency should sustain to provide U_i with S_k . It is strictly dependent on both the welfare system and the service typology. As an example, in the Italian welfare system, in most cases, a citizen must partially contribute to the provided service and, generally, his contribution is directly proportional to his yearly income. As a consequence, the cost a government agency must sustain for providing a service is higher for citizens having a lower income¹.

In our experiments we have considered the Italian welfare system and we have asked an expert to define tables specifying the cost of services into consideration for various income ranges. Some of these tables can be found at the address <http://www.ing.unirc.it/ursino/ismis06/tables.html>.

Generally, the budget B_k that a government agency can associate with S_k is much smaller than the sum of the costs necessary to supply S_k to all citizens; as a consequence, it is necessary a decision making activity to identify the subset US_k of users (called *target community* in the following) that can gain the maximum benefit from the activation of S_k .

PAA determines US_k by solving the following optimization problem:

$$\begin{aligned} \max \quad & \sum_{i=1}^{|USet|} b_{ik} x_i \\ \text{s.t.} \quad & \sum_{i=1}^{|USet|} c_{ik} x_i \leq B_k \\ & x_i \in \{0, 1\}, 1 \leq i \leq |USet| \end{aligned}$$

Here $USet$ represents the set of possible users; x_i is a variable associated with the user U_i ; it is set to 1 if and only if the system assumes that U_i belongs to US_k ; it is set to 0 otherwise.

The previously defined problem is the well-known *0/1 Knapsack Problem*. It has been widely studied in the past and has been successfully applied in disparate contexts. It is a well-known $\mathcal{NP}$ -hard problem but various polynomial heuristics have been proposed for solving it (see [5] for all details about the Knapsack Problem and its heuristics).

The solution of the previous Knapsack Problem allows managers of the government agency to determine the set of users that could gain the maximum benefit from the activation of S_k . At this point the characteristics of the target community can be examined to establish if the benefit of S_k is sufficient to justify its activation.

¹ It is worth pointing out that, generally, also the benefit of a service is higher for citizens having a lower income (see below).

2.5 Discussion

In our system the exploitation of the Intelligent Agent technology provides various benefits w.r.t. a classical distributed system based on cooperating processes. In fact, all properties typical of the Intelligent Agent technology are profitably exploited: autonomy allows our system to use the knowledge of both user profiles and service features for producing suggestions to both government agencies' managers and citizens, without requiring the human intervention; proactivity allows it to act on behalf of citizens to identify the most interesting services for them; adaptivity (and the corresponding learning capability) allows it to construct and handle user profiles and to adapt its behaviour to them; collaborative behaviour makes involved agents to continuously exchange messages and strictly cooperate to pursue the goal of producing suggestions to both citizens and government agencies' managers.

3 Experimental Results

The prototype of our system has been realized in JADE (Java Agent DEvelopment Framework), a FIPA (Foundation for Intelligent Physical Agents) compliant platform for developing Intelligent Agents. In our system the Public Administration Agents and the Service Manager Agents operate on Personal Computers equipped with a 3.4 GHz CPU and 1 Gb of RAM, whereas User Profile Agents run on a Personal Computer with a 2.6 GHz CPU and 256 Mb of RAM.

In our tests we have considered a set $USet = \{U_1, U_2, \dots, U_{200}\}$ of 200 users. Their distribution, against their yearly income and age, can be found at the address <http://www.ing.unirc.it/ursino/ismis06/tables.html>. In addition, we have considered a set $SSet$ of 40 services, relative to various domains, such as Health, Social Security, Education, Public Transports, and so on; some information about their main features and their activation budgets has been extracted from Italian Central Government Web site <http://www.italia.gov.it>; the other necessary information has been partially provided by a human expert. Their distributions, against their reference domain and activation budget, can be found at the address <http://www.ing.unirc.it/ursino/ismis06/tables.html>. Finally, we have considered a set $BSet$ of possible activation budgets; an activation budget represents the money (expressed in Euros) that a government agency has at disposal for offering a service for one month. The set $BSet$ we have considered was: $BSet = \{15000, 30000, 45000, 60000, 75000, 90000, 105000, 120000, 135000, 150000\}$.

In order to evaluate the quality of the suggestions produced by our system, we have applied the following framework:

- For each service $S_k \in SSet$:
 - We have applied our system to identify the Set of Users $US_k \subseteq USet$ who might obtain the maximum benefit from the activation of S_k .
 - We have asked a domain expert to specify the Set of Users $US_k^* \subseteq USet$ who, in his opinion, might gain the maximum benefit from the activation of S_k .

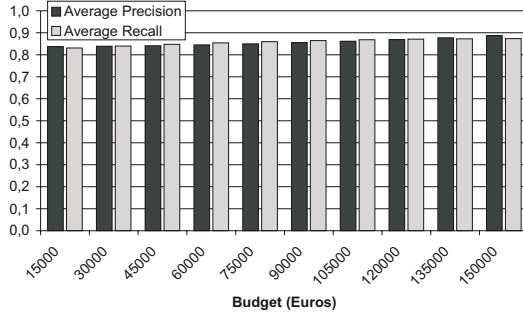


Fig. 2. Average accuracy measures for various values of the activation budget

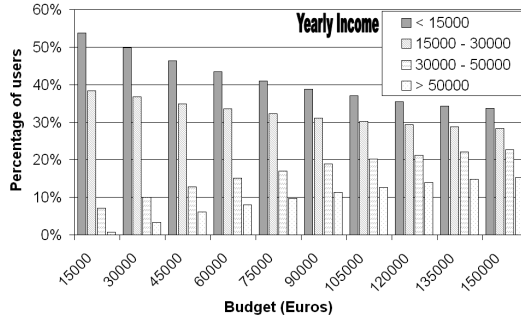


Fig. 3. Percentages of citizens selected by our system for a service

- We have computed some accuracy measures relative to S_k ; specifically, the measures we have adopted are: (i) *Precision* (hereafter Pre_k), indicating the share of users correctly selected by the system among those it proposed; it is defined as $Pre_k = \frac{|US_k^* \cap US_k|}{|US_k|}$; (ii) *Recall* (hereafter Rec_k), denoting the share of users correctly selected by the system among those the expert provided; it is defined as $Rec_k = \frac{|US_k^* \cap US_k|}{|US_k^*|}$. Precision and Recall fall within the real interval $[0, 1]$; clearly, high values of Precision and Recall, indicate a good system accuracy.
- We have averaged our accuracy measures for all services relative to the various budgets. Specifically, for each budget $B_p \in BSet$: (i) we have considered the set $SSet_p \subseteq SSet$ of services such that each of them can be activated with a budget equal to B_p ; (ii) we have averaged the accuracy measures obtained for all services of $SSet_p$.

The values of the average accuracy measures for each budget of $BSet$ are reported in Figure 2. From the analysis of this figure we may observe that average measures are substantially constant and quite high. The high values of accuracy measures show that our system is really capable of simulating the behaviour and

the reasoning of the human expert and, therefore, that it is really adequate of supporting decision makers of government agencies.

The constant values of accuracy measures might be justified by the following reasoning: generally, in presence of scarce resources, a welfare system tends to privilege those citizens having a low income; in fact, even if the costs of activating a service for them are higher, the corresponding benefits are much higher. As a consequence, any human expert would tend to privilege this category of citizens in presence of low budgets; the high values of Precision and Recall for low activation budgets show that the algorithms underlying our system are really capable of simulating this welfare systems' trend. When the available budget increases, both the human expert and our system tend to choose, for a service, also citizens with a higher income; as a consequence, there is no substantial change of Average Precision and Average Recall.

In order to verify the correctness of this reasoning, we have performed a further test; specifically, for each budget of *BSet*, we have computed the percentage of citizens, selected by our system, having a yearly income less than 15000 Euros (resp., between 15000 and 30000 Euros, between 30000 and 50000 Euros, higher than 50000 Euros). The obtained results are shown in Figure 3.

From the analysis of this figure we can observe that, for low budgets, our system really privileges low-income citizens; in fact, the percentage of low-income citizens proposed for a service is higher than the corresponding percentage in *USet* (see <http://www.ing.unirc.it/ursino/ismis06/tables.html>). When the available budget increases, our system begins to select also citizens with higher incomes. Finally, when the available budget is very high, our system has at disposal a lot of resources and, consequently, it can propose also high-income citizens for a service; as a consequence, the citizen distribution (against their income) selected for a service tends to coincide with the corresponding citizen distribution in *USet*.

4 Conclusions

In this paper we have presented a multi-agent system for supporting managers of government agencies in their decision making activities. We have pointed out that the proposed system is an attempt of applying the Intelligent Agent technology in the context of e-government. With regard to this, we have shown that many features typical of agents (such as learning capability) can play a key role in this context.

As for future research efforts in this context, we plan to define a framework for evaluating the effects of composing two or more existing services in such a way to obtain a service package. Moreover, we would like to refine our approach by taking into account the "life scenario" associated with a user in a certain time. In fact, it could happen that the activation of a new service (e.g., the activation of a new bus line in a city) might be very useful for citizens in a certain life scenario (e.g., during the working days), whereas it might be not relevant in other scenarios (e.g., during the week-end).

Acknowledgments

The authors thank Matteo Camera for his contribution to the implementation of the proposed approach.

References

1. Accenture Consulting. e-Government Leadership Rhetoric vs. Reality—Closing the Gap. Technical report, Available at www.accenture.com/xdoc/en/industries/government/final.pdf, 2001.
2. A. Fay. A fuzzy knowledge-based system for railway traffic control. *Engineering Applications of Artificial Intelligence*, 13(6):719–729, 2000.
3. K. Goto and Y. Kambayashi. A new passenger support system for public transport using mobile database access. In *Proc. of the International Conference on Very Large Databases (VLDB 2002)*, pages 908–919, Hong Kong, China, 2002. VLDB Endowment.
4. D. Gouscos, G. Mentzas, and P. Georgiadis. PASSPORT, a novel architectural model for the provision of seamless cross-border e-government services. In *Proc. of the International Workshop on Database and Expert Systems Applications (DEXA 2001)*, pages 318–322, Munich, Germany, 2001. IEEE Computer Society.
5. H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.
6. A. Kobsa. Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11:49–63, 2001.
7. G. Koutrika and Y. Ioannidis. Personalized queries under a generalized preference model. In *Proc. of the IEEE International Conference on Data Engineering (ICDE 2005)*, pages 841–852, Tokyo, Japan, 2005. IEEE Computer Society Press.
8. W. Lam, S. Mukhopadhyay, J. Mostafa, and M. Palakal. Detection of shifts in user interests for personalized information filtering. In *Proc. of ACM International Conference on Research and Development in Information Retrieval (SIGIR '96)*, pages 317–325, Zurich, Switzerland, 1996. ACM Press.
9. G. Marchionini, H. Samet, and L. Brandt. Introduction to the special issue on Digital Government. *Communications of the ACM*, 46(1):24–27, 2003.
10. J. Sharrard, J.C. McCarthy, M.J. Tavilla, and J. Stanley. Sizing US government. Technical report, Forrester Research, Inc., Cambridge, Massachusetts, USA, 2000.

A Butler Agent for Personalized House Control

Berardina De Carolis, Giovanni Cozzolongo, and Sebastiano Pizzutilo

Dipartimento di Informatica – University of Bari
www.di.uniba.it

Abstract. This paper illustrates our work concerning the development of an agent-based architecture for the control of a smart home environment. In particular, we focus the description on a particular component of the system: the Butler Interactor Agent (BIA). BIA has the role of mediating between the agents controlling environment devices and the user. As any good butler, it is able to observe and learn about users preferences but it leaves to its “owner” the last word on critical decisions. This is possible by employing user and context modeling techniques in order to provide a dynamic adaptation of the interaction with the environment according to the vision of ambient intelligence. Moreover, in order to support trust, this agent is able to adapt its autonomy on the basis of the received user delegation.

1 Introduction

In a past project¹, we developed *C@sa* an agent-based organization for controlling a smart home environment. In particular, we focused our research on organizing these agents in such a way that they could allow for an easy adaptation of the environment behavior to the needs of its inhabitants [1].

In its current implementation, *C@sa* uses the following classes of agents: **Context Agents**, that are used for providing information about context features (i.e. temperature, light level, humidity, etc.), **Operator Agents**, controlling devices, **Supervisors**, controlling and coordinating behaviors of Operator Agents belonging to the same *influence sphere*, and an **HouseKeeper** that acts as a facilitator [2] since it knows all the agents that are active in the house and also the goal they are able to fulfill.

In this paper we will focus our discussion on the description of our work concerning the interaction level with the smart home environment.

Applying Ambient Intelligence (AmI) solutions, to handle the control of house behavior, may help in making the house services fruition easy, natural and adapted to the user needs [3]. To this aim, as far as interaction is concerned, the following approaches have been proposed:

- i) **agent mediated interaction**, in which a personal agent assists the user by hiding the complexity of difficult tasks, automating repetitive tasks, training or teaching the user, etc..

¹ National Project: PRIN03: “*L’ambiente domestico informatizzato: progetto e verifica dell’integrazione di utente, tecnologia e prodotto*”- Università di Bari and Siena, Politecnico di Milano.

- ii) **explicit interaction with the artifacts in the house**, in this case the user interacts explicitly with a device by changing, for instance, its state. This can be done directly through the device switches or through a more natural control interface accessible for instance using a touch screen display or a vocal input or by allowing the user to manipulate artifacts according to the so called tangible interface paradigm [4].
- iii) **implicit interaction through sensors**, in this case, the interaction is triggered by sensor data obtained from the user and other observations in the world (e.g. temperature, light conditions). In both cases, an “agent behind the scene” might be the counterpart of interaction. If needed, such an invisible agent can also emerge on-stage offering a direct interface to the user, e.g. to clarify some potential misunderstanding.

These approaches can be mixed by providing the right interaction metaphor according to the user and context situation. To this aim, we have developed an agent, called **Butler Interactor Agent (BIA)**, dedicated to the facilitation of the interaction between the agents that control the automation of house devices and the user. When the personal agent interaction metaphor is reported to the house, the intelligent assistant becomes a virtual butler to which the user may issue commands or delegate some tasks. We decided to give this role to this personal agent since for several centuries the butler character, with its typical features, such as loyalty, discretion and general helpfulness, has been present in the families of all the world as symbol of cooperation [5].

The **BIA** has been designed then as a kind of user assistant and provides an interface between the house and its inhabitants according to the Ambient Intelligent vision [6]. As any good butler, it should be always present when needed but not too intrusive, it should be able to learn about users preferences but it should leave to its “owner” the last word on critical decisions [7]. To this aim it employs user and context modeling for supporting proactive adaptation of the interaction with the house. Moreover, in order to support trust, this agent is able to adapt its autonomy on the basis of the user authorization level.

As far as user modeling is concerned, this agent takes into account elements of uncertainty and the possibility to deal with incomplete information that is related with the dynamicity of the interaction typical of an intelligent environment. In this context, in fact, it is important to infer user’s goals from his/her actions, interaction history, context and implicit and/or explicit feedback that the user provides as a consequence of the system actions. These motivations, together with the possibility to represent in a compact way probabilistic information, lead us to model the BIA’s reasoning as a Belief Network (BN) [8]. As we will show later on, it employs this probabilistic approach for modeling user preferences and needs and, by logging user actions in the house, learns variation in the probabilistic distribution of the model.

The paper is structured as follows. In the next Section, we describe the BIA in details focusing on its user modeling capabilities and on its autonomy. In particular, in Section 3 we provide an example of its application and evaluation in the context of house automation. Conclusions and future work directions are illustrated in Section 4.

2 The Butler Interactor Agent

As mentioned in the Introduction, we decided to adopt the butler metaphor for interacting with the house. According to [9], "Personal agents are personalized interface agents, whose fundamental component is a user model, or personal profile, which they employ when undertaking their task. Profiles allow personal agents to perform tasks according to the needs and preferences of the user, turning the computer into an intelligent personal assistant". Then user modeling is a crucial task for the effectiveness and efficacy of the BIA agent. In particular, when modeling the user in a dynamic context, such as the one described in this work, it is important to update the initial default model dynamically and consistently according to the user actions that may be interpreted as a positive or negative feedback toward the decision taken by the agent.

We will show how the user model is built and managed. Then we will describe how the user's feedback influences the agent's autonomy.

2.1 Agent Based User Modeling

Historically, user models have been used to enable the adaptation and personalization of services or information presentation to users (e.g.[10][11]). However, when applying user modeling in ambient intelligence it is necessary to relate user preferences to the context so as to allow the environment to answer in an adaptive and proactive manner to the possible user needs.

Moreover, due to the dynamicity typical of this application domain it is important to understand how the model has to evolve in such a way as to learn and update contextual information about the user [12]. Thus, machine learning is considered as a practical method to learn and represent a user's interests, preferences, knowledge, goals, habits, etc. in order to adapt the services to the user's individual characteristics [13].

On the other hand, an agent has to show some properties that are considered prerequisites for its intelligence: it needs to be reactive to changes in the environment, goal-oriented and proactive in taking the initiative to achieve this goal, and it should have a certain level of autonomy to complete its tasks without intervention of the user [7]. For these reasons, in order to support proactive house behavior and to meet user's expectations adopting an agent based approach to user modeling seems appropriate.

If we reason in agent-related terms we should define then which are the *percepts*, the *reasoning* and the *effectors* of this agent [14]. In this case the perceptual input of this agent is related to context features, interaction history and explicit user actions or statements. Obviously the effectors will be messages to be passed to the agent controlling a given influence sphere in order to render these decision as house actions.

The inference reasoning behavior will include two aspects: an **engine to reason on user** presences and needs in the current context and a **learning** technique to maintain the intrinsic variability of this domain.

As far as the reasoning engine is concerned, unobtrusiveness and context awareness typical of this domain involve capturing and making sense of imprecise and sometimes conflicting data and uncertain physical worlds. Bayesian networks are a way to deal with reasoning in this type of application domains since they are a

powerful way of handling uncertainty, especially when there are causal relationships between various events. They are useful for performing probabilistic information fusion and higher-level reasoning on contextual situations.

However, one limitation of predefined adaptation models is that the user must re-configure the system in order to reflect changes in his/her preferences. Performing such reconfigurations could be a frustrating or annoying task for the user. However, by observing the behavior of the user an intelligent environment could potentially learn variations in the inference rules of the model in order to provide dynamic adaptations. One potential problem that may arise from proactive modeling-based adaptation is that the user might not agree with the agent's decisions. In this case, according to the level of autonomy, the agent may explicitly ask the user whether the decided action has to be performed or not, refining its reasoning accordingly. In this way, a trust relationship between the BIA and the user can be established.

In order to test and validate our approach, before building the model for all the possible influence spheres that are related with our daily life, we decided to focus on the modeling of the comfort influence sphere. Working on the modeling of this type of service seems appropriate for testing our approach since comfort involves several devices and, according to several definitions, its perception is highly individual and involves different human senses (temperature, light, intimacy, sounds, level of noise, smell, etc.)[15].

The main components involved in our simulation experiment and their interaction with each other are illustrated in Figure 1.

Contextual factors to be considered in the experiment are internal and external temperature, humidity, noise level, light level, the user's activity (working, housekeeping, relaxing, etc), the status of house appliances involved in the termic comfort influence sphere (window, air conditioning, biometric user data).

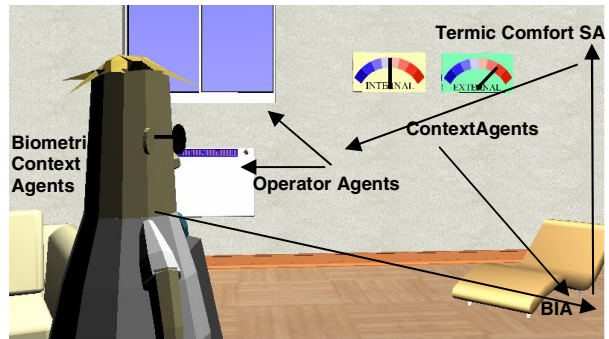


Fig. 1. Context-B opased Intelligent Environment Control

The BIA starts from an initial user model built learning from statistical data after the collection of preferences data in the considered domain. Then it instantiates this model for the specific user by copying it as user profile. From that moment it collects information about contexts, user and house actions as an interaction history. Next, it induces variation on the probability distribution of the initial user profile according to the weights that the agent gives to each user feedback action. Using this model and according to its level of autonomy, our agent can provide a suggestion to the user or perform automatically an action when the physical environment changes. Finally, whenever the system does provide a proactive adaptation, the user can override the adaptation by providing an explicit feedback.

2.1.1 Building the Initial Model

Since, at this stage of the project, we do not have a real domotic house with sensors to collect data for building the initial user model structure to be used by the BIA, we collected a set of data about termic comfort habits. In particular, we have collected diaries of 20 people, living in a house with the following devices: air conditioning, heating, windows. Their distribution is summarized in Table1.

In writing the diary we asked these people to contextualize their needs and preferences to their activity, attitude, room, and other factors such as daytime, humidity, temperature, wind, etc..

In total we collected an initial dataset of about 200 entries to be given as input to a bayesian network learning tool.

An example of diary entry is the following:

Sunday, 5.00 p.m , spring, working on the computer, inside is getting hot, I'm feeling a bit hot, outside is cooler and slightly windy, I open the window.

The function we assign to our user model is to enable the BIA to infer the desirable action to execute in relation to the context situation. We applied the Bayesian network structure learning algorithm of Hugin 6.5².

Table 1. Distribution of people involved in diary compilation

Sex	Male	10
	Female	10
Age	<25	2
	25-35	5
	35-45	6
	45-55	4
	>55	3
Role	Father	4
	Mother	4
	Child	4
	Single	8

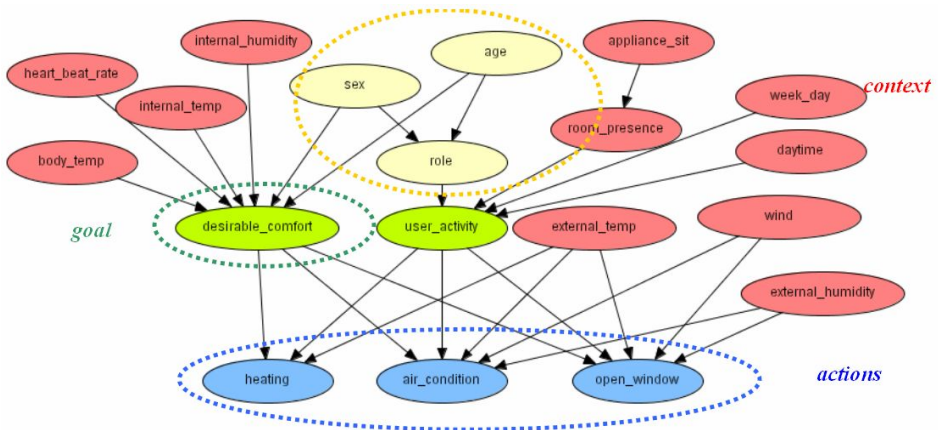


Fig. 2. Initial user and context model

In our case, introduced variables belong to the following categories :

- Stable and known user characteristics: sex, age and role;
- Dynamically evolving, unknown user features: desirable comfort, user activity;

² www.huginexpert.com

- c. *'Observable' context features*: temperature, internal temperature, wind, humidity, heart beat rate, etc.
- d. *Desirable action*: open windows, turning on the air condition, turning on the heating system;

In this network, **root** nodes are mainly related to context factors and to stable and known information about the user. Then we have a set of **intermediate** nodes that are used to infer information about the user, for instance his/her presumed goal about termic comfort, his/her activity, etc. The leaves nodes correspond to hidden variables that are used to infer which are the most probable actions that the environment should perform for achieving the triggered goal.

2.2 BIA's Autonomy

Another important issue regards the level of autonomy that a personal agent has in making its decision about what is the right thing to do for satisfying the user needs in a given context [7]. The BIA agent has a certain level of autonomy when achieving its intentions, in particular in designing our agent we are interested in modeling the: i) **Execution Autonomy**: related to execution of actions (tasks, subtasks, request of services, and so on), ii) **Communication Autonomy**: related to the level of intrusiveness in communicating to the user. The Execution Autonomy is inversely proportional to the Communication one.

Each dimension has an associated value in a 5 values scale from "null" to "high". The "null" value represents the absence of autonomy, and the BIA has to execute only the actions explicitly requested by the user. It cannot infer any information or decide to modify task execution without explicitly asking it to the user. The opposite value, "high", represents the maximum level of autonomy that gives to the agent the chance to decide what to do according to constraints imposed by the user. The other values represent a "medium" level of autonomy that is inversely proportional to the certainty of the agent's decision. Each level is associated to a threshold. When the autonomy level is high the probability of an action is not considered (the threshold is 0). The threshold for other levels rises with an increment of 0.25. The null value of the autonomy level implies that the threshold is 1, so that the agent has to ask the confirmation for every action.

Autonomy values are revised according to the type of feedback the user provides to the agent: *positive* feedback enforces the autonomy on that category of task, *negative* one reduces it. This simple mechanism aims at implementing a relation between the agent's level of autonomy and the type of user delegation. Initially, the user sets explicitly the autonomy level for a task and can always revise it.

If the level of autonomy is high the BIA executes the actions, sending messages to the OAs, without considering its likelihood. When the user disagrees, performing an action which is contrary to the last action decided by the BIA, the level of autonomy is decreased. The context situation and the action performed by the user are logged. Each contrary action decreases the autonomy for that kind of task of one point. When the autonomy level becomes null the agent will ask the user for confirming every action. If the user confirm the action then the level increases, and consequently the entry is added to the log. All the feedback actions of the user will be used to adjust the model as we will see in the next section.

4 Learning Individual Contextual Models: An Application Example

The BIA knowledge base is then composed by a set of *belief about the user*, a set of *beliefs about the context*, that are updated according to what the context agents perceive, and a set of general **goals** that are related to the influence spheres defined in the smart home environment (i.e. termic comfort, wellness, security, etc.).

According to the model that we have seen in Figure 2, on the basis of the current beliefs, BIA general goals are specialized for the current context situation according to what has been inferred by its probabilistic reasoning. Each goal is related to a set of possible actions belonging to a given influence sphere. These possible actions will be triggered according to the inferred probability by sending messages to other agents controlling the house or to the user according to the agent's autonomy.

In order to describe how the agent reasoning works we will show an example in the following scenario. Lets' suppose that:

- the user is *working* on the computer,
- inside for some reasons is getting *hot* and the perceived temperature on the user body is also getting *higher*,
- outside is *cold* and slightly *windy*,
- the level of execution autonomy set by the user for the termic comfort task is *high*.

When there is at least one change in a context value the involved Context Agent, controlling physical sensors, sends a context-changed message to the BIA (i.e. if the physical situation changes from "normal" to "hot"). Then, the BIA infers goals to achieve in the current situation and consequently the most appropriate action based on the probability of the states of the relative variables present in the belief network. In the considered scenario, since outside is cooler the BIA will infer that the desirable user comfort is very likely to be *to have a cooler temperature*.

In the described context the most probable action to do will be to open the window and, according to the autonomy level, this action will be executed without asking to the user.

As we described in the previous Section, when an action is accepted or not by the user (by doing an opposite action or another one for achieving the same goal), an entry corresponding to the description of the context and the action, will be used to reinforce (in case of positive feedback) or to weaken (in case of negative one) the likelihood of executing that action in that context, thus improving the accuracy of the model. Since a feedback has a consequence also on the autonomy level of the agent, this will be decreased or increased accordingly.

The log will collect all these feedback actions and could be used to revise the model offline to investigate the motivation for the errors.

Let's suppose now that the user has some papers on the desk and therefore will close the windows to avoid them to fly away. This consequent negative feedback will be used to update the model by using this contextual change for learning variation in the probability distribution of the network. The following experience (Table2) will be also added to the log.

Table 2. Example of entry for describing the “close the windows” feedback

wind	open_ window	air_ condition	heating	body_ temp	desirable_ comfort	user_ activity	external_ temp	internal_ temp
true	false	false	N/A	hot	cooler	working/ studying	mild	hot

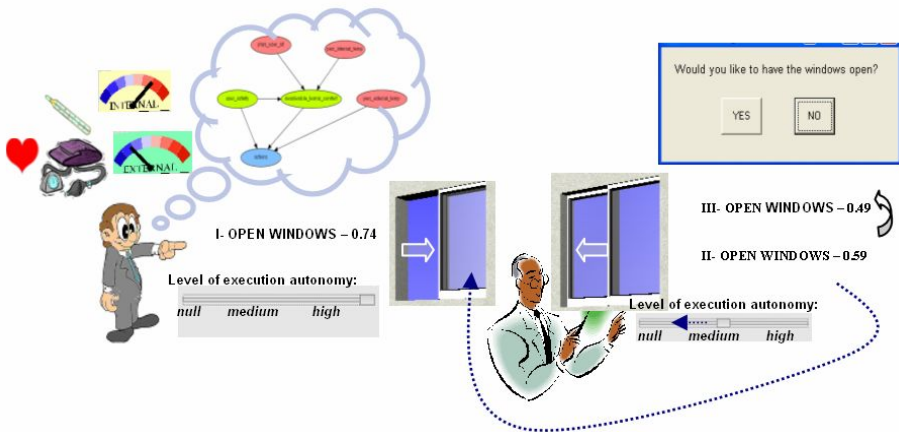


Fig. 3. The described scenario

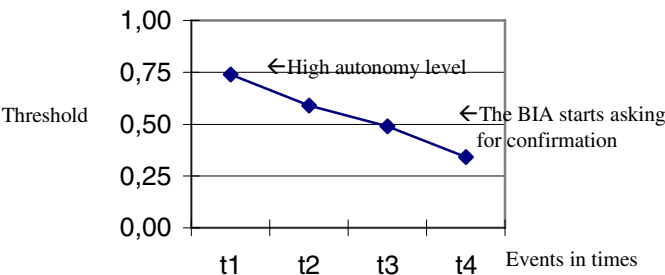


Fig. 4. The variation of likelihood of opening the window for the described scenario

The probability of opening the windows decreases from the initial value 0.74 to 0.59 (Figure 3), and also the autonomy is decreased accordingly to level 4. When the same situation occurs the probability of the action is higher than the updated threshold (0.25), so the agent doesn’t ask the confirmation before opening the windows again. The user disagrees, close the windows the second time and the probability of the action decreases to 0.59 as well as the autonomy to level 3. The probability is now under the actual threshold (0.75) so the agent asks for the user confirmation to perform the action (Figure 4). Every time the user disagrees with the BIA the situation is logged and the probability continues to decrease, as well as the level of autonomy.

The model is update continuously so after some times the user will agree and let the agent perform the action. In this case the autonomy rises for that kind of task. Every time an action is performed and the user doesn’t give negative feedback, this

reinforces the model, and the probability of the right action for that context rises. In our case after three times the BIA will infer the right action to do.

The new learned behavior is considered right until the user disagrees with them. For example, when the user will stop working, and there are no more papers on the desk, the BIA will not open the windows in the context described before. Then a periodically revision of the model is necessary.

Obviously, in the same scenario, if the autonomy is initially low the system will ask confirmation until the level increases to the higher value. This level is always adjustable by the user, who can disable the automatic variation of the autonomy level.

We conducted an experiment to determine if the methodology described in this paper would be useful in anticipating the user needs. We asked to 25 users to interact with a 3D simulation of 3 interaction scenarios and to express their feedback about the actions decided by the BIA agent. According to their evaluation, the learned model for this experiment inferred correctly the right action in 92% cases. We are aware that in interacting with a real smart home environment this level of accuracy may drop, however, the results of this experiment encourage us to continue our experiments with a full set of sensors, actuators, and multi-target functions.

4 Conclusions and Future Work Directions

This paper has explored the potential of utilizing a Butler Interactor Agent for mediating between the agents controlling the home devices and the user. BIA utilizes user and context models, formalized as a belief network, for inferring actions to be executed in the smart home environment in order to satisfy user goals. The initial structure of the user model has been learned from data extracted by a collection of user daily diaries. Then, we built a prototype system for evaluating the performance of the learning capability of the BIA agent. Positive and negative user feedback was employed to learn variation on the probability distributions of the model. In order to support user trust and control over the agent behavior, we gave to our agent a level of execution and communication autonomy that is set by the user for families of tasks to be executed in the house. The BIA autonomy then varies during the interaction according to the type of user feedback. Negative feedback decrease autonomy, positive ones increase it if not otherwise specified.

At this stage of our experiment, we are only considering the state of the internal comfort as a target function. However, since this approach is general enough to be applied to other influence spheres we intend to consider multi-target functions for the next step in our experiment.

We believe that context history has a concrete role for supporting proactive adaptation in smart environments. Indeed by analyzing the context log it could be possible to understand that in the initial structure of the model some factors were not considered.

In the near future, we intend to investigate on this issue and the implications of giving a “body” to the BIA agent and we hope to continue our experiments with a full set of sensors, actuators, and multi-target functions.

References

1. De Carolis, B., Cozzolongo, G., Pizzutilo, S., Plantamura, V. L. Agent-Based Home Simulation and Control. ISMIS 2005: 404-412.
2. Open Agent Architecture: <http://www.ai.sri.com/~cheyer/papers/aai/oaa.html>
3. Shadbolt, N. Ambient Intelligence. IEEE Intelligent Systems. (Vol.18, No.4) July/August 2003.
4. Ishii, H. and Ullmer, B., Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms, in Proceedings of Conference on Human Factors in Computing Systems CHI '97), (Atlanta, March 1997), ACM Press, pp. 234-241.
5. Bartnek, C., eMuu – An Embodied Emotional Character for the Ambient Intelligent Home, Technische Universiteit Eindhoven, 2002
6. Grill, T., Kotsis, G., Ibrahim. I.K., Agents for ambient intelligence: Support or nuisance. Journal of the "Österreichische Gesellschaft für Artificial Intelligence" OGAI, 23(1): ISSN 0254-4326 (2004). 19–26,
7. Falcone, R., Castelfranchi, C., Tuning the collaboration level with autonomous agents: A principled theory; LNAI 2175: AI*IA 2001: Advances in Artificial Intelligence. Springer. (2001). 212-224
8. Pearl, J., Belief networks revisited. Artificial Intelligence, 59:, (1993). 49-56
9. Downes, S., <http://www.downes.ca/cgi-bin/page.cgi?post=165>
10. Fink, J. and Kobsa, A., User Modeling for Personalized City Tours. Artificial Intelligence Review, 18(1), (2002). 33-74
11. Byun, H.E., Cheverst k., Exploiting User Models and Context-Awareness to Support Personal Daily Activities, Workshop in UM2001 on User Modelling for Context-Aware Applications, Sonthofen, Germany. (2001)
12. Rao, S., Cook, D.J., Predicting Inhabitant Actions Using Action and Task Models with Application to Smart Homes, International Journal of Artificial Intelligence Tools, 13(1), (2004). 81-100,
13. Mitchell, T. M., Machine Learning, McGraw-Hill (1997)
14. Russell, S.J., Norvig, P., Artificial Intelligence: a Modern Approach. Prentice Hall. (1998).
15. Spronk, B., A House is not a Home: Witold Rybczynski Explores the History of Domestic Comfort. Aurora Online, 2001.

Incremental Aggregation on Multiple Continuous Queries

Chun Jin and Jaime Carbonell

Language Technologies Institute, School of Computer Science
Carnegie Mellon University, Pittsburgh, PA 15213 USA
{cjin, jgc}@cs.cmu.edu

Abstract. Continuously monitoring large-scale aggregates over data streams is important for many stream processing applications, e.g. collaborative intelligence analysis, and presents new challenges to data management systems. The first challenge is to efficiently generate the updated aggregate values and provide the new results to users after new tuples arrive. We implemented an incremental aggregation mechanism for doing so for arbitrary algebraic aggregate functions including user-defined ones by keeping up-to-date finite data summaries. The second challenge is to construct shared query evaluation plans to support large-scale queries effectively. Since multiple query optimization is NP-complete and the queries generally arrive asynchronously, we apply an incremental sharing approach to obtain the shared plans that perform reasonably well. The system is built as a part of ARGUS, a stream processing system atop of a DBMS. The evaluation study shows that our approaches are effective and efficient on typical collaborative intelligence analysis data and queries.

1 Introduction

Aggregates are standard database operations and are widely used in data warehousing applications. They have been studied and commercialized with great success. However, recently, the emergence of the data stream processing presents new challenges to compute multiple aggregates over ever-changing data streams.

Consider a collaborative environment for intelligence analysis. An analyst wants to quickly respond to emergency situations, and discover unusual developments of special events from structured data streams. For example, an analyst concerning terrorism activities wants to detect signals of bio-terrorism attacks. So he sets up a set of continuous aggregate queries on the stream of hospital patient records to monitor possible outbreaks of contagious diseases that are possibly transmitted by bio-attack agents.

Analysts with overlapping interests and expertise share resources, results, and opinions to collaborate complex strategic and tactic tasks. In a collaborative environment, multiple analysts may set up different monitoring queries at different times. New queries are formulated when new patterns are identified as worth tracking by analysts or automatic novel intelligence tools. We expect probably hundreds or thousands of such queries to run concurrently.

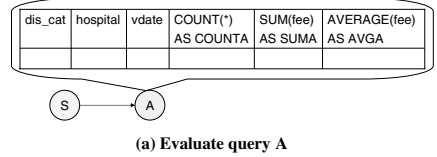
Further, the wide varieties and complexity of tasks suggest the mighty richness and complexity of expected aggregate queries, including the extensive applications of complex user-defined aggregates, such as TrackClusterCenters. Tracking cluster centers is

an important novel intelligence tool [10] to monitor trends and event evolutions where tuples are viewed as points in a n -dimension feature space and clustered, and the cluster centers are updated when new points arrive.

Many aggregate functions, *algebraic functions*, including MIN, MAX, COUNT, SUM, AVERAGE, STDDEV, and TrackClusterCenters, can be incrementally updated upon data changes without revisiting the entire history of grouping elements; while other aggregates, *holistic functions*, e.g. quantiles, MODE, and RANK, can not be done this way. We implemented an efficient incremental aggregation mechanism for arbitrary algebraic aggregate functions including user-defined ones. The system also supports holistic functions by reaggregating the stream history.

Consider a query A monitoring the number of visits and the average charging fees on each disease category in a hospital everyday, shown in Fig 1(a). When new tuples of patient records from the stream S arrive, the aggregates $COUNT(*)$ and $AVERAGE(fee)$ can be incrementally updated if $COUNT(*)$ and $SUM(fee)$ are stored. Our system will generate a query plan (query network), shown in Fig 2(a), to efficiently update the aggregate results.

```
SELECT  dis_cat, hospital, vdate,
        COUNT(*), AVERAGE(fee)
FROM    S
GROUP BY CAT(disease) AS dis_cat
        hospital,
        DAY(visit_date) AS vdate
(a) Query A
```



```
SELECT  hospital, vdate,
        AVERAGE(fee)
FROM    S
GROUP BY hospital,
        DAY(visit_date) AS vdate
(b) Query B
```

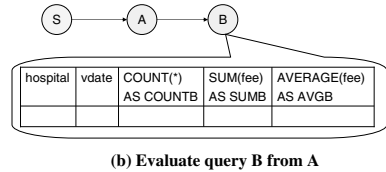


Fig. 1. Two Query Examples: A and B

Fig. 2. Evaluating Queries A and B

Constructing the shared evaluation plan among multiple queries is necessary to support large-scale queries effectively. Since multiple query optimization (MQO), even on the simple case of aggregating over single columns[7], is NP-complete[21,2,20,16], the global optimization on large-scale queries is impractical. Further, queries generally arrive intermittently in real applications, such as in intelligence monitoring. To make it tractable, a natural solution is incrementally adding new queries to obtain a reasonably good plan within a reasonable time. To do that, the system needs to record the computations of the existing query plan R . Given a new query Q , the system searches the common computations between Q and R , chooses the optimal sharing path, and expand R with new computations to obtain Q 's final results.

Assume R contains just query A . Now consider that a new query B arrives. It monitors the daily average fees in a hospital, shown in Fig 1(b). B groups can be obtained by compressing A groups on the $CAT(disease)$ dimension. Further, B 's aggregates can be obtained from A as well. Thus the system shares A 's results to evaluate B , as shown in Fig 2(b). This sharing process is called *vertical expansion* in this paper.

We implemented two sharing strategies. The first one is to choose the optimal sharing node when there are multiple sharable ones. And the second is *rerouting*. After a new node B is created, the system checks if any existing nodes can be improved by being evaluated from B . These nodes are disconnected from their original parents and connected to B by vertical expansion. If the two queries A and B mentioned above arrive in the reverse order, the sharing can still be achieved by applying rerouting.

The system is built as a part of ARGUS[15,14], a stream processing system atop of a DBMS (Oracle) for immediate practical stream processing functionalities to existing database applications. Besides aggregations, ARGUS also supports selections, joins, set operators, and view references with the incremental evaluation and sharing approaches.

The evaluation study shows that the execution time of incremental aggregation grows much slower than the naive regrouping method as the aggregation size increases, and is up to 10 times faster in our experiments. The study also shows that the incremental sharing provides significant improvement on large-scale queries, and that vertical expansion provides significant improvement on typical data increment sizes.

2 Related Work

Efficient aggregations have been studied widely for data warehousing applications. The CUBE[11], ROLLUP, and GROUPING SETS[2,13,19] generate multiple aggregates by grouping on different sets of aggregating columns (dimensions), and are implemented in commercial DBMS's. These operators allow the query optimizer to share computations among multiple aggregates. [7] proposed a hill-climbing approach to search for efficient shared plans on a large number of GROUPING SETS queries. [22] showed how to choose finer-granularity yet-not-requested aggregates to be shared by multiple aggregates to improve computation efficiency. While these OLAP-oriented approaches consider sharing among multiple aggregates, they assume all queries are available at the time and perform the optimization as an one-shot operation. However, many stream applications, e.g. intelligence monitoring, request registering intermittently arrived queries on an one-by-one basis. Our system accommodates this by recording and searching existing computations to incrementally construct the query plan.

Our work is also relevant to view-based query optimization [9] in terms of searching common computations. However, the optimization only involves identifying the sharable computations without dynamic construction of new computations, and do not concern with incremental aggregations upon data changes. While automatic view maintenance is related to incremental aggregation, previous works[4][12] focus on selection-join queries, not on aggregate queries.

Aggregations over data streams have also been studied. Window aggregates[17] explore stream/query time constraints to detect the aggregating groups that no longer grow and thus avoid unnecessary tuple buffering. Continuously estimating holistic functions,

i.e. quantiles and frequent items, in distributed streams [8,18] is another important problem. Since incremental aggregation is not applicable, the approaches focus on approximate techniques and models to bound errors and to minimize communication cost. Orthogonal to these efforts, our work focuses on incremental aggregations for general algebraic functions and support large-scale query systems.

Recent stream processing systems, such as STREAM[3], TelegraphCQ[5], and Aurora[1], are general-purpose Data Stream Management Systems focusing on system-level problems, such as storage, scheduling, approximate techniques, query output requirements, and window join methods. To our knowledge, however, techniques toward incremental aggregation for large-scale queries are not discussed. Our work is orthogonal to these efforts and can be applied as part of the optimizer on such systems.

NiagaraCQ[6] is the closest system to ours. It monitors large-scale continuous queries with incremental query evaluation methods, and registers new queries incrementally with periodical regrouping optimization. However, the system focuses on queries with joins and selections, not on aggregate functions.

3 System Design

In this section, we present the system architecture, incremental aggregation, vertical expansion, and sharing strategies.

3.1 System Architecture and System Catalog

The system builds a shared query network. Each network node, presenting a data stream S , is associated with two tables, the *historical table* S_H to materialize the historical data part, and the *temporary table* S_N to materialize the new data part which will be flushed and appended to the historical table later. At any time, $S = S_H \cup S_N$. An aggregate query may have a GROUPBY clause to specify a set of grouping expressions which is called *dimension set* $\mathbb{D}$. A group value is denoted as $g^{\mathbb{D}}$.

Incremental aggregation on an aggregate node is realized by PL/SQL code blocks instantiated from code templates. The whole query network evaluation is realized by a set of stored procedures that wrap up the code blocks in the order the nodes appear in the network. The stored procedures run periodically on the new data, and incrementally update aggregate results.

Fig 3 and 4 show the system architecture and the system catalog. *System catalog* is a set of system relations to record the topological and node-specific information of the shared query network to allow common computation search, network expansion, and reoptimization. *GroupExprIndex* records all canonicalized group expressions[14], *GroupExprSet* records the dimension sets, and *GroupTopology* records the topological connections of the aggregate nodes. Each dimension set has a *GroupID*, is uniquely associated with an aggregate node in *GroupTopology*, and contains one or more *GroupExpressions* recorded in *GroupExprIndex*. Each node has a unique entry in *GroupTopology* which records the node name, its direct parent (the node from which the results are computed), its original stream table, and the *GroupID*. *GroupColumns* records the projected group expressions and aggregate functions for each node.

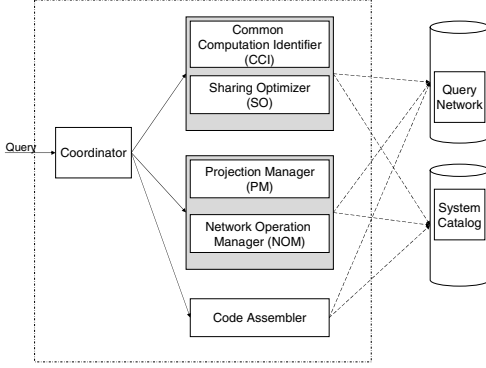


Fig. 3. System Architecture

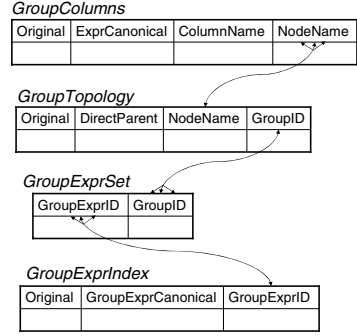


Fig. 4. System Catalog

Given a new query B , the system takes a few steps to expands the existing query network R , so R also evaluates B . First, the common computation identifier identifies the common computations between B and R . Particularly, it identifies all dimension sets $\{\mathbb{D}_A\}$ that are supersets of $\mathbb{D}_B$, identifies the nodes $\{A\}$ associated with $\{\mathbb{D}_A\}$, and then checks whether the nodes in $\{A\}$ contain all columns needed for query B . Second, the sharing optimizer chooses the optimal sharing path, particularly the optimal node A for vertical expansion. Third, the network operation manager and the projection manager expand R and record the changes in the system catalog. And finally, the code assembler assembles code blocks into executable stored procedures.

3.2 Incremental Aggregation

Aggregate functions can be classified into three categories[11]:

Distributive: Aggregate function F is distributive if there is a function G such that $F(S) = G(F(S_j)|j = 1, \dots, K)$. COUNT, MIN, MAX, SUM are distributive.

Algebraic: Aggregate function F is algebraic if there is a function G and a multiple-valued function F' such that $F(S) = G(\{F'(S_j)\}|j = 1, \dots, K)$. AVERAGE is algebraic with $F'(S_j) = (SUM(S_j), COUNT(S_j))$ and $G(F'(S_j)|j = 1, \dots, K) = \frac{\sum F'_1(S_j)}{\sum F'_2(S_j)}$ where F'_i is F' 's i th value.

Holistic: Aggregate function F is holistic if there is no constant bound on storage for describing F' . Quantiles are holistic.

Distributive and algebraic functions can be incrementally updated with finite data statistics while holistic functions can not. In this paper, we also refer distributive functions as algebraic, since they are special cases of algebraic functions.

To perform incremental aggregation for arbitrary algebraic functions including user-defined ones, we use two system catalog tables to record the types of the necessary statistics and the updating rules, shown in Tables 1 and 2. Table *AggreBasics* records the necessary bookkeeping statistics for each algebraic function. Argument X in

BasicStatistics indicates the exact match when binding with the actual value while *W* indicates the wild-card match. Table *AggreRules* records incremental aggregation rules. The rule of a distributive function specifies how the new aggregate is computed from S_H and S_N ; and the rule of an algebraic function specifies how the new aggregate is computed from the basic statistics.

Incremental aggregation is shown in Algorithm 1 and is illustrated by Fig 5 on *AVERAGE(fee)* for query *A*. Algorithm 1 also shows the time complexity T_i of each step. The merge step is realized with a hash join on A_H and A_N with $T_{hash2} = O(|A_H| + |A_N|)$. If the A_H hash is precomputed and maintained in RAM or on disk with perfect prefetching, the time complexity is $T_{prefetch2} = O(|A_N|)$. The duplicate-drop step is realized by a set difference $A_H - A_N$, which can be achieved by hashing with $T_{hash4} = O(|A_H| + |A_N|)$ or by prefetched hashing with $T_{prefetch4} = O(|A_N|)$. However, we observed that it took $T_{curr4} = O(|A_N| * (|A_N^H|)) = O(|A_N|^2)$ on the DBMS, where $|A_N^H|$ is the number of the groups in S_H to be dropped. If the incremental aggregation is implemented in a DBMS or a DSMS as a built-in operator, both merge and duplicate-drop steps can be achieved with hashing. With the prefetching, the complexity will be linear to $|A_N|$. Therefore, the time complexities on the current implementation, build-in operator with hashing, and with prefetch are following:

$$T_{curr} = O(|S_N^2| + |A_H|), T_{built-in} = O(|S_N| + |A_H|), \text{ and } T_{prefetch} = O(|S_N|).$$

Algorithm 1. Incremental Aggregation

0. PredUpdate State. A_H contains update-to-date aggregates on S_H .

1. Aggregate S_N , and put results into A_N . $T_1 = O(|S_N|)$

2. Merge groups in A_H to A_N . $T_{hash2} = O(|A_H| + |A_N|)$, $T_{prefetch2} = O(|A_N|)$

3. Compute algebraic aggregates in A_N from basic statistics
(omitted for distributive functions). $T_3 = O(|A_N|)$

4. Drop duplicates in A_H that have been merged into A_N .

$$T_{curr4} = O(|A_N| * |A_N^H|) = O(|A_N|^2),$$

$$T_{hash4} = O(|A_H| + |A_N|), T_{prefetch4} = O(|A_N|)$$

5. Insert new results from A_N to A_H , preferably after A_N has been sent to the users.

$$T_5 = O(|A_N|)$$

Fig 6 shows the procedure to instantiate the incremental aggregation code for function *AVERAGE(fee)* in query *A*. 1. The function is parsed to obtain the function name and the list of the actual arguments. 2. The basic statistics and updating rules are retrieved. 3. The statistics are parsed and their formal arguments are substituted by the actual arguments. 4. The statistics are renamed and stored in *GroupColumns*. 5. The name mapping is constructed based on above information. 6. The updating rules are instantiated by substituting formal arguments with the renamed columns.

Table 1. AggreRules

AGGREGATE FUNCTION	AGGREGATE CATEGORY	INCREMENTAL AGGREGATION RULE	VERTICAL EXPANSION RULE
AVERAGE	A	$SUM X / COUNT W$	$SUM X / COUNT W$
SUM	D	$SUM X(H) + SUM X(N)$	$SUM(SUM X)$
MEDIAN	H	NULL	NULL
COUNT	D	$COUNT W(H) + COUNT W(N)$	$SUM(COUNT W)$

Table 2. AggreBasics

AGGREGATE FUNCTION	BASIC STATISTICS	BASIC STATID
AVERAGE	$COUNT(W)$	$COUNT W$
AVERAGE	$SUM(X)$	$SUM X$
SUM	$SUM(X)$	$SUM X$
COUNT	$COUNT(W)$	$COUNT W$

3.3 Vertical Expansion and Sharing Strategies

Vertical expansion is not applicable to holistic queries. However, holistic queries can still be shared if they share the same dimension sets. In following discussion, we focus on sharing among distributive and algebraic functions.

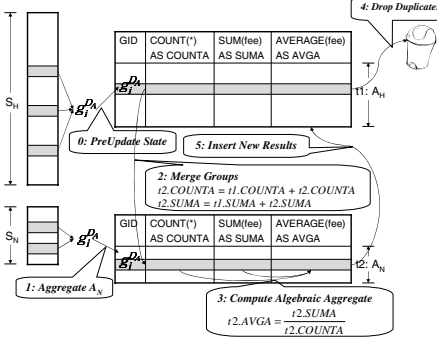


Fig. 5. Incremental Aggregation

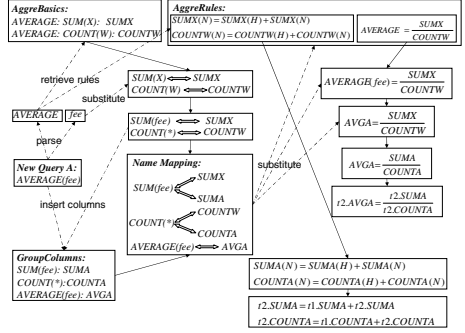


Fig. 6. Incremental Aggregation Instantiation

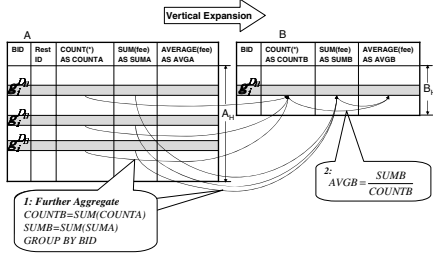


Fig. 7. Vertical Expansion

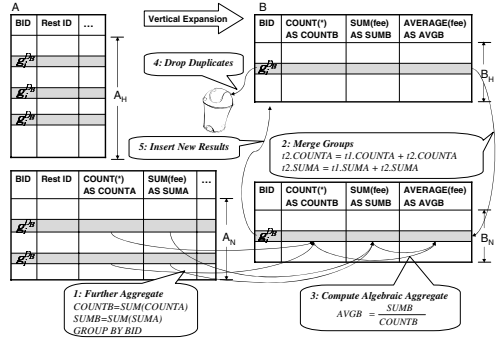


Fig. 8. Incremental Aggregation on Vertical Expansion

Section 1 showed that the query B can be evaluated from the query A (see Fig 2). Fig 7 shows how the vertical expansion creates and initiates B from A . The process is comprised of two steps and takes time of $T^{VInit} = O(|A_H|)$. The further-aggregate step executes the code instantiated from the vertical expansion rules stored in the *AggreRules*, and the algebraic-computing step is applicable only to algebraic functions.

Fig 8 shows the incremental aggregation for $AVERAGE(fee)$ in query B . It is the same to the procedure shown in Fig 5 except the first step which performs further aggregation from A_N instead of from S_N . The time complexities are following:

$T_{curr}^V = O(|A_N^2| + |B_H|)$, $T_{built-in}^V = O(|A_N| + |B_H|)$, and $T_{prefetch}^V = O(|A_N|)$.

Given the new query B , there may be multiple nodes from which a vertical expansion can be performed. According to the time complexity analysis, the optimal choice is the node A such that $|A_H|$ is the smallest. If A does not contain all aggregate functions or bookkeeping statistics needed by query B , a *horizontal expansion* is performed to add necessary aggregate functions to A .

After the new node B is created, the system invokes the rerouting procedure. It checks if any existing node C can be sped up by being evaluated from B . We apply a simple cost model to decide such rerouting nodes. If 1. B contains all the aggregate functions needed by C , and 2. $|B_H| < |P_H^C|$ where P^C is C 's current parent node, then C will be rerouted to B . The system applies a simple pruning heuristic. If a node C satisfies both conditions, and a set of nodes $\{C_i\}$ satisfying the first condition are descendants of C , then any node in $\{C_i\}$ should not be rerouted, and so are dropped from consideration.

4 Evaluation Study

We conduct experiments to show: 1. effect of incremental aggregation (*IA*) vs. the naive reaggregation approach (*NIA*), 2. effect of shared incremental aggregation (*SIA*) vs. unshared incremental aggregation (*NS-IA*), 3. performance changes of vertical expansion (*VE*) and non-vertical expansion (*NVE*) with regard to $|S_N|$. The experiments were conducted on an HP PC computer with Pentium(R) 4 CPU 3.00GHz and 1G RAM, running Windows XP.

Two databases are used. One is the synthesized FedWire money transfer transaction database (Fed) with 500000 records. And the other is the Massachusetts hospital patient admission and discharge record database (Med) with 835890 records. Both databases have a single stream with timestamp attributes. To simulate the streams, we take earlier parts of the data as historical data, and simulate the arrivals of new data incrementally. Table 3 shows the historical and new data part sizes used in the experiments.

Table 3. Evaluation Data

	Fed	Med
$ S_H $	300000	600000
$ S_N $	4000	4000

Table 4. Total execution time
in seconds

	Fed 350Q	Med 450Q
IA	662	316
NIA	6236	938

Table 5. Vertical expansion
statistics

	Pair1	Pair2
$ S_H $	300000	300000
$ A_H $	95050	94895
$ B_H $	10000	10000

We use 350 queries on Fed and 450 queries on Med. These queries are generated systematically. Interesting queries arising from applications are formulated manually. Then more queries are generated by varying the parameters of the seed queries. Some of these queries aggregate on selection and self-join results.

Fig 9 shows the execution times of *IA* and *NIA* on each single query, and the ratio between them, NIA/IA . A NIA/IA ratio above 1 indicates better *IA* performance. Since

there are more fluctuations on diversified Med queries, we show running averages over 20 consecutive queries in Fig 9(b) to make the plot clear. The queries are sorted in the increasing order of $AggreSize = |S_H(A)| * |A_H|$, shown on the second Y axis. $|S_H(A)|$ is the actual aggregation data size of the historical part after possible selections and joins. We experimented with two other metrics, $|S_H(A)|$, $|S_H(A)| + |A_H|$, which show less consistent trends to the time growth and the NIA/IA ratio. This indicates that the DBMS aggregation operator on S_H has time complexity of $|S_H(A)| * |A_H|$.

$AggreSize$ indicates the query characteristics. There are about 250 queries in both Fed and Med whose $|S_H(A)|$ is 0, shown to the left of the vertical cut lines and are sorted by the NIA/IA ratio. Unsurprisingly, their execution times are very small. The small time fluctuations are caused by the DBMS file caching.

For the remaining queries, incremental aggregation is better than non-incremental aggregation. Particularly, it gains more significant improvements when the aggregation size is large. These large-size queries dominate the execution time in multiple-query systems. Thus significant improvements on such queries are significant to the whole system performance, as shown in Table 4.

Fig 10 shows the total execution times of SIA and $NS-IA$ by scaling over the number of queries. Clearly, incremental sharing provides improvements, particularly on Fed where queries share more overlap computations.

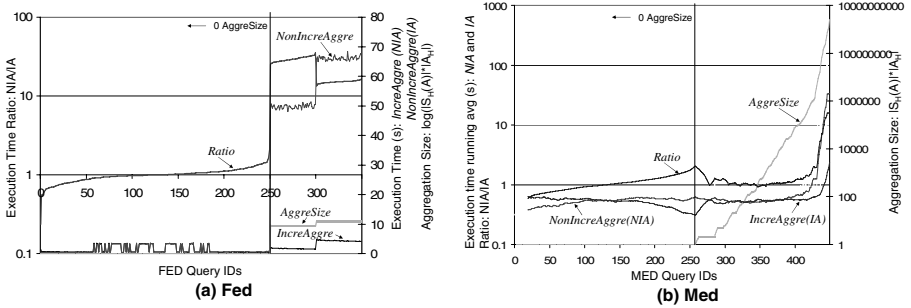


Fig. 9. Sharing

To study how the incremental size $|S_N|$ influences the vertical expansion performance, we select two Fed query pairs and compare vertical expanded plans (VE) and non-vertical expanded plans ($NonVE$) on them by varying the incremental sizes from 1 to 30000 tuples in exponential scale, as shown on the X axis in Fig 11. In each query pair, one query B can be shared by vertical expansion from another query A , and their data sizes are shown in Table 5. Fig 11 shows two types of execution times, IBT and ITT . IBT is the time to update the whole incremental batch S_N , indicated on the left Y axis, and ITT is the average time to update an individual tuple in each incremental batch, $ITT = IBT / |S_N|$, indicated on the right Y axis. The VE performance gets close to NVE as $|S_N|$ gets larger, since $|S_N|^2$ in T_{curr} becomes dominant and dims the VE advantage. We expect that this situation can be avoided with the hasing implementation.

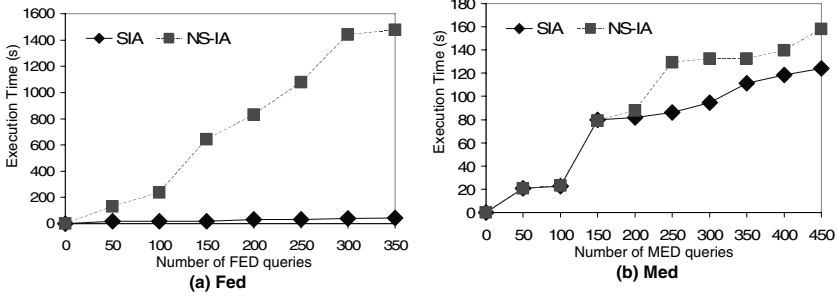


Fig. 10. Sharing

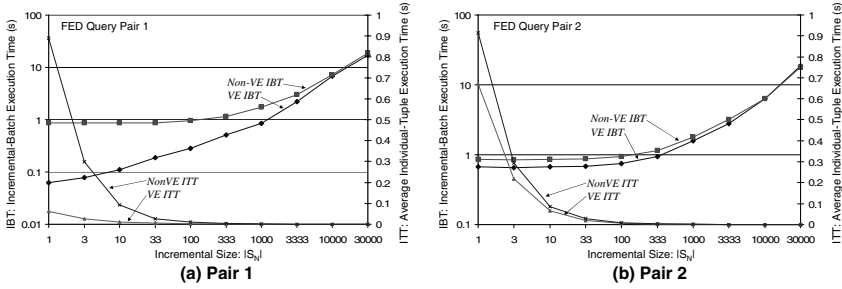


Fig. 11. Vertical expansion

5 Conclusion and Future Work

We implemented a prototype system that supports incremental aggregation and incremental sharing for efficient monitoring of large-scale aggregates over data streams. It also supports incremental aggregation on user-defined functions, which are critical to real-world applications. The evaluation showed that the incremental aggregation improves the performance significantly on dominant queries, and so the incremental sharing over a large number of queries, and that vertical expansion is effective in typical data increment sizes. We plan to migrate the prototype onto an open-source DSMS and implement the algorithms as built-in operators.

Acknowledgements. This work was supported in part by ARDA, NIMD program under contract NMA401-02-C-0033. The views and conclusions are those of the authors, not of the U.S. government or its agencies. We thank Christopher Olston, Phil Hayes, Santosh Ananthraman, Bob Frederking, Eugene Fink, Dwight Dietrich, Ganesh Mani, and Johny Mathew for helpful discussions.

References

1. D. J. Abadi and et al. Aurora: a new model and architecture for data stream management. *VLDB J.*, 12(2):120–139, 2003.
2. S. Agarwal and et al. On the computation of multidimensional aggregates. In *VLDB*, pages 506–521, 1996.
3. B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *PODS*, pages 1–16, 2002.
4. J. A. Blakeley, N. Coburn, and P.-Å. Larson. Updating derived relations: Detecting irrelevant and autonomously computable updates. *ACM Trans. Database Syst.*, 14(3):369–400, 1989.
5. S. Chandrasekaran and et al. TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. In *CIDR*, January, 2003.
6. J. Chen, D. J. DeWitt, F. Tian, and Y. Wang. Niagaracq: A scalable continuous query system for internet databases. In *SIGMOD Conference*, pages 379–390, 2000.
7. Z. Chen and V. R. Narasayya. Efficient computation of multiple group by queries. In *SIGMOD Conference*, pages 263–274, 2005.
8. G. Cormode and et al. Holistic aggregates in a networked world: Distributed tracking of approximate quantiles. In *SIGMOD Conference*, pages 25–36, 2005.
9. D. DeHaan, P.-Å. Larson, and J. Zhou. Stacked indexed views in Microsoft SQL Server. In *SIGMOD Conference*, pages 179–190, 2005.
10. C. Gazen, J. Carbonell, and P. Hayes. Novelty Detection in Data Streams: A Small Step Towards Anticipating Strategic Surprise. In *NIMD PI Meeting*, Washington, DC, 2005.
11. J. Gray and et al. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *J. Data Mining and Knowledge Discovery*, 1(1):29–53, 1997.
12. A. Gupta, H. V. Jagadish, and I. S. Mumick. Data integration using self-maintainable views. In *EDBT*, pages 140–144, 1996.
13. V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. In *SIGMOD Conference*, pages 205–216, 1996.
14. C. Jin and J. Carbonell. Toward Incremental Sharing On Continuous Queries. Tech. Report available upon request from authors, Carnegie Mellon Univ, 2005.
15. C. Jin, J. Carbonell, and P. Hayes. ARGUS: Rete + DBMS = Efficient Continuous Profile Matching on Large-Volume Data Streams. In *ISMIS*, pages 142–151, 2005.
16. A. Y. Levy, A. O. Mendelzon, Y. Sagiv, and D. Srivastava. Answering queries using views. In *PODS*, pages 95–104, 1995.
17. J. Li, D. Maier, K. Tufte, V. Papadimos, and P. A. Tucker. Semantics and evaluation techniques for window aggregates in data streams. In *SIGMOD Conf*, pages 311–322, 2005.
18. C. Olston, J. Jiang, and J. Widom. Adaptive filters for continuous queries over distributed data streams. In *SIGMOD Conference*, pages 563–574, 2003.
19. K. A. Ross and D. Srivastava. Fast computation of sparse datacubes. In *VLDB*, pages 116–125, 1997.
20. W. Scheufele and G. Moerkotte. On the complexity of generating optimal plans with cross products. In *PODS*, pages 238–248, 1997.
21. T. K. Sellis and S. Ghosh. On the multiple-query optimization problem. *IEEE Trans. Knowl. Data Eng.*, 2(2):262–266, 1990.
22. M. Zhang, B. Kao, D. W.-L. Cheung, and K. Yip. Mining periodic patterns with gap requirement from sequences. In *SIGMOD Conference*, 2005.

Location-Aware Multi-agent Based Intelligent Services in Home Networks*

Minsoo Lee, Yong Kim, Yoonsik Uhm, Zion Hwang, Gwanyeon Kim,
Sehyun Park**, and Ohyoung Song

School of Electrical and Electronics Engineering, Chung-Ang University,
221, Heukseok-Dong, Dongjak-Gu, Seoul 156-756, Korea
{lemins, ykim, neocharisma, zhwang, cityhero}@wm.cau.ac.kr,
{shpark, song}@cau.ac.kr

Abstract. The development of intelligent multi-agent systems involves a number of concerns, including mobility, context-awareness, reasoning and mining. Towards ubiquitous intelligence this area of research addresses the intersection between mobile agents, heterogeneous networks, and ubiquitous intelligence. This paper presents a development of hardware and software systems to address the combination of these interests as Location-Aware Service. Our architecture performs the intelligent services to meet the respective requirements. By adding autonomous mobility to the agents, the system becomes more able to dynamically localize around areas of interest and adapt to changes in the ubiquitous intelligence landscape. We also analyze some lessons learned based on our experience in using location-aware multi-agent techniques and methods.

Keywords: ubiquitous networks, location-awareness, multi-agent, context aware system, Ontology, smart home, pervasive computing.

1 Introduction

Context-aware computing refers to an information system which has the special capability to sense and react to dynamic environments and activities. Context-aware computing systems including software, hardware, networks are becoming overly complex with ever increasing scale and heterogeneity. To cope with the ubiquitous complexity, the concern with autonomous agents has been growing which are capable of computing, communicating, and behaving smartly with some intelligence. One of the profound implications of such autonomous smart agents is that various kinds and levels of intelligence will cooperate with everyday objects, environments, systems and ourselves. As solutions for context-aware

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the Chung-Ang University HNRC(Home Network Research Center)-ITRC support program supervised by the IITA(Institute of Information Technology Assessment).

** The corresponding author.

system some of the recent studies have focused on the secure context management framework [1], the database-oriented [2], the agent-oriented [3] and the service-oriented approach [4]. However, the high mobility of the users generates new research challenges because of the heterogeneities of access technologies, network architectures, protocols and various Quality of Service (QoS) demands of mobile users [5]. Furthermore, in the development of home network systems for a smart home the problem of efficient Location-Aware Service (LAS) management [6] for the highly intelligent services has not been adequately considered in respect of autonomous services as well as satisfying the QoS guarantees.

Location-aware computing can be regarded as the essential elements of the ubiquitous intelligence [7]. Location-aware computing is made possible by the location sensing, wireless communication, and mobile computing systems. Distributed location sensors in wireless networks offer many new capabilities of agents for contextually monitoring and service environments [8]. By making such location-aware systems, we can increase application-space with the higher intelligence for the home networks mainly by providing dynamic context-dependent deployment, seamless services, automatic QoS adaptation [9][10].

In this paper we propose a location-aware service architecture based on an Ontology Based Context-Aware Service Agent System (OCASS) architecture [11] that performs the intelligent context-aware services to meet the respective requirements. We utilized the designed context model (HomeNet Ontologies) using the Web Ontology Language (OWL) for providing facility when the system performs to sense, mine and reason about context. By adding autonomous mobility support to the agents, the system becomes more able to dynamically localize around areas of interest and adapt to changes in the ubiquitous intelligence landscape. Performance evaluations and lessons learned are also presented to demonstrate the effectiveness of our architecture.

The rest of this paper is organized as follow. Section 2 gives related works about location-aware computing towards a smart world. In section 3, we describe our architecture, implementation testbed and performance evaluations. Finally, we conclude in Section 4.

2 Motivations and Related Works

Recent research work of context-aware systems [2][3][4] mainly use the several initial contexts that are sensed by context providers [12], so they may lead to lack of more intelligent services for recent moving users who request extremely various needs according to the specific spot in a smart space. One of the central issues in a smart home is location-awareness based on ubiquitous intelligence paradigm, whose goal is to provide fine-grained persistent services following the roaming path of a moving user. Location is a crucial component of context, and much research in the past decade has focused on location-sensing technologies including WiFi-based, ultrasound, RFID, vision-based [13], the middleware support [14]. However, intelligent location-aware services in the future smart home networks brings about many challenges. The problem of location-aware efficient

resource management has not been considered adequately in respect of enhancing the communication performance as well as satisfying the QoS guarantees. In a smart home, many kinds of location sensors may be available. The task of making sense of this vast amount of sometimes contradictory information, known as *sensor fusion*, presents a major challenge. Inferencing and learning methods for processing data from Hierarchical Hidden Markov Model [15], various location sensors, Bayes filters, and particle filters are developed principled methods of incorporating sensor uncertainty. Our work builds on efforts by several other results of frameworks and applications [6][14][16][17], location measurement derived from multiple sensors [18][19]. The main advantage of LAS agents comes from the intelligent localization. If the number of location information requests during a service is high, so many data transmission for inferencing and mining can be significantly avoided by using LAS agents.

3 Location-Aware System in Home Environment

In this section, we describe our architecture, hardware and software implementation and performance evaluations.

3.1 Location-Aware Service Agent System Architecture

We designed LAS agents with OCASS architecture as shown in Fig. 1 for a smart home to meet the location-aware computing requirements.

* *Location-Aware Service Agent.* For location-aware intelligent services we deploy LAS Agents with location tracking and prediction schemes in each room of our smart home. *Location Tracking Agent* takes charge of user location tracking events coming from the location sensors. *Location Decision & Prediction Agent* selects the right spot and entity (with service capabilities) closest to the user. In the case of home environments a user's path may be effectively predicted along the corridors and doors by the location history based algorithms [20] that has a record of the previous user movements and take into account the respective probability of movements together with factors such as direction and speed [21].

* *Knowledge Repository.* *Knowledge Repository* provides various contexts and information, structured and unstructured data, rules. Also, it stores static and dynamic patterns as knowledge. Using RuleML [22] and JESS[23] script language, rule-based engine accesses and stores, deletes rules. *HomeNet Ontologies* [11] are the inside domain ontologies.

* *Context Managing Agent.* *Context Managing Agent* has a function of collecting, filtering and managing contexts. We implemented a part of this *Context Managing Agent* related OWL using Jena[24].

* *Interface Agent.* *Interface Agent* performs representing of contextual data from MMI (Multimodal Interaction framework)[25] or sensor agents, creating queries, transforming and transmitting the result about the queries.

* *Inference & Mining Agent.* *Ontology Reasoner* is responsible for checking class consistency and implied relationship and asserting inter-ontology relations.

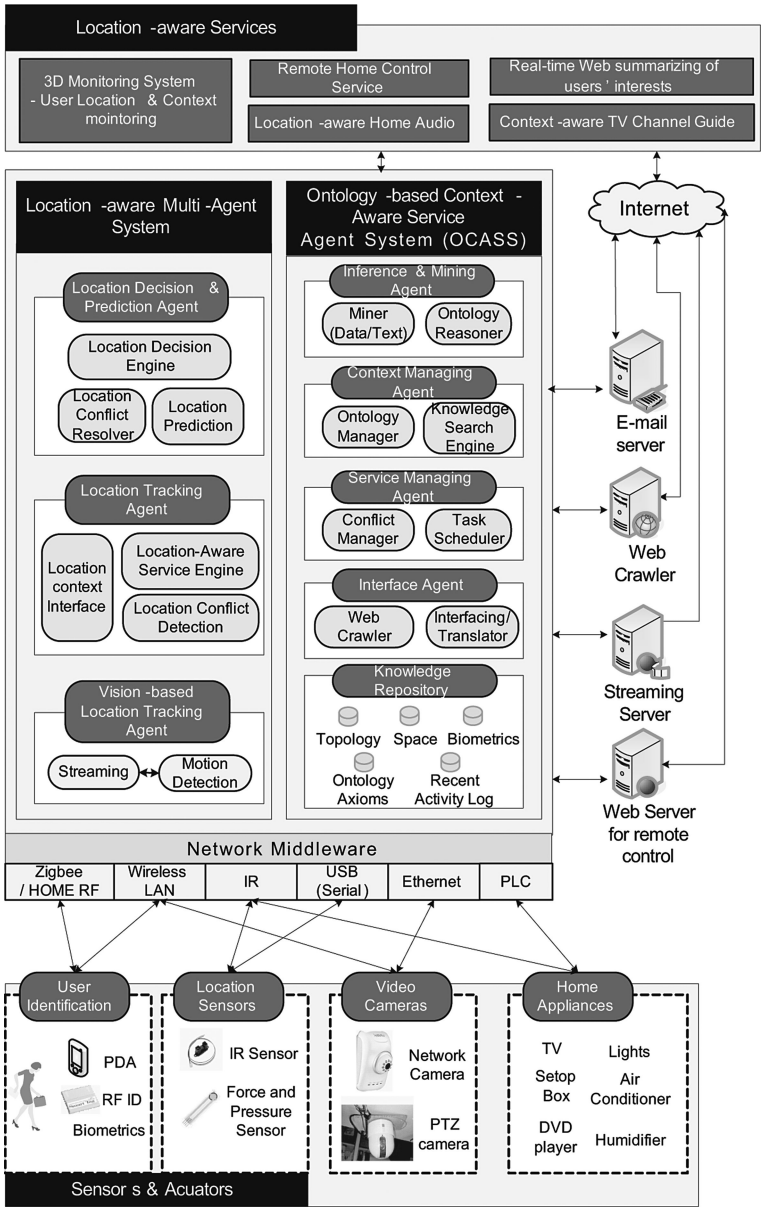


Fig. 1. Location-Aware Service Agents with our Ontology-based Context-Aware Service Agent System(OCASS)

It is implemented using a rule-based reasoning engine. *Miner* generates contexts and patterns to fit a user's interest and design. *Miner* monitors the history of contexts that *Reasoner* already ordered *Context Managing Agent* to acquire to

use for making statistic information. We implemented the learning mechanism of *Data Miner* using Hierarchical Hidden Markov Model (HHMM) [15].

* *Service Managing Agent*. *Service Managing Agent* administrates services including the modules to provide context aware services.

3.2 System Implementation and Intelligent Services in Smart Home

We deployed our system in an actual home setting and improved the location evaluation function to perform well in that environment. Fig. 2 shows the floor map of the real target area (Home Network Research Center) and Fig. 3 shows the deployment of our location-aware information push service. The overall area is about 162 square meters. Although our system design is not sophisticated enough to work in different environments, it does work well in our target home. To cover the large space 46 IR long distance measuring sensors and 13 pressure sensors were used. We deployed network cameras with motion detection and alarm function. The detected images can be remotely saved in the *Vision-based Location Tracking Agent* and then retrieved as needed for the location of a user. To cover the living room we also deployed the vision-based location tracking

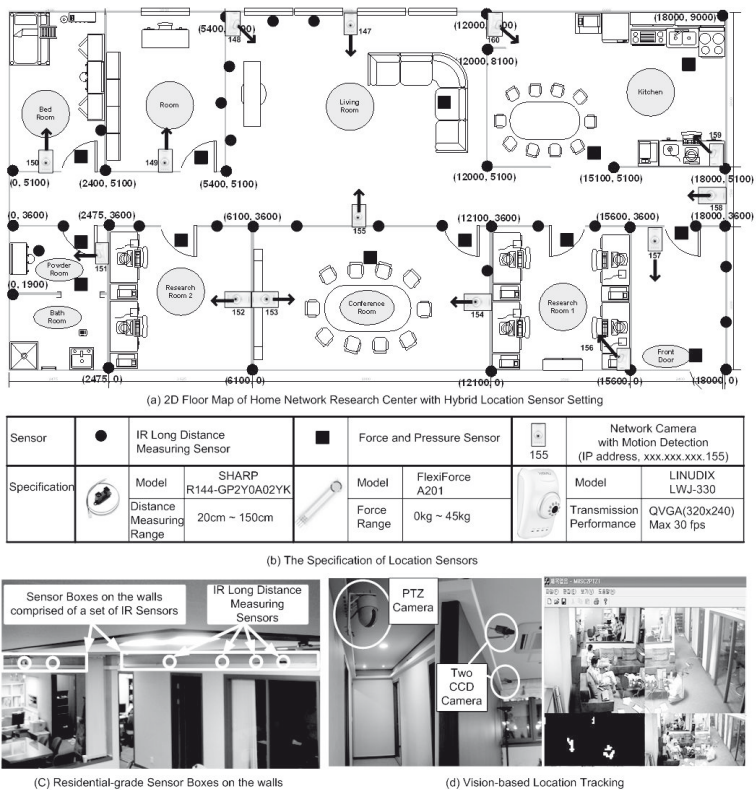


Fig. 2. Floor Map of Home Network Research Center with multiple Location Sensors

system which consists of CCD cameras and a pan-tilt-zoom (PTZ) camera with feature fusion-based people tracking algorithm [13]. LAS agents run on Pentium 4 1.4MHz machines with IEEE 802.11b Wireless Local Area Networks (WLANs) cards. OCASS gateway was designed based on Intel Pentium 4 2GHz with a JVM from the Java 2 Enterprise Edition. Agents open a remote-method-invocation (RMI) connection to send requests to the appropriate service agents. We also used Jena2 [24] as the OWL parser. As the *Reasoner*, we modified the rule based engine JESS to be able to interact other agents. IR generators controls home appliances by converting Power Line Communication (PLC) signal into IR signal. As the terminal devices, we used WLAN enabled Compaq iPAQ PDAs.

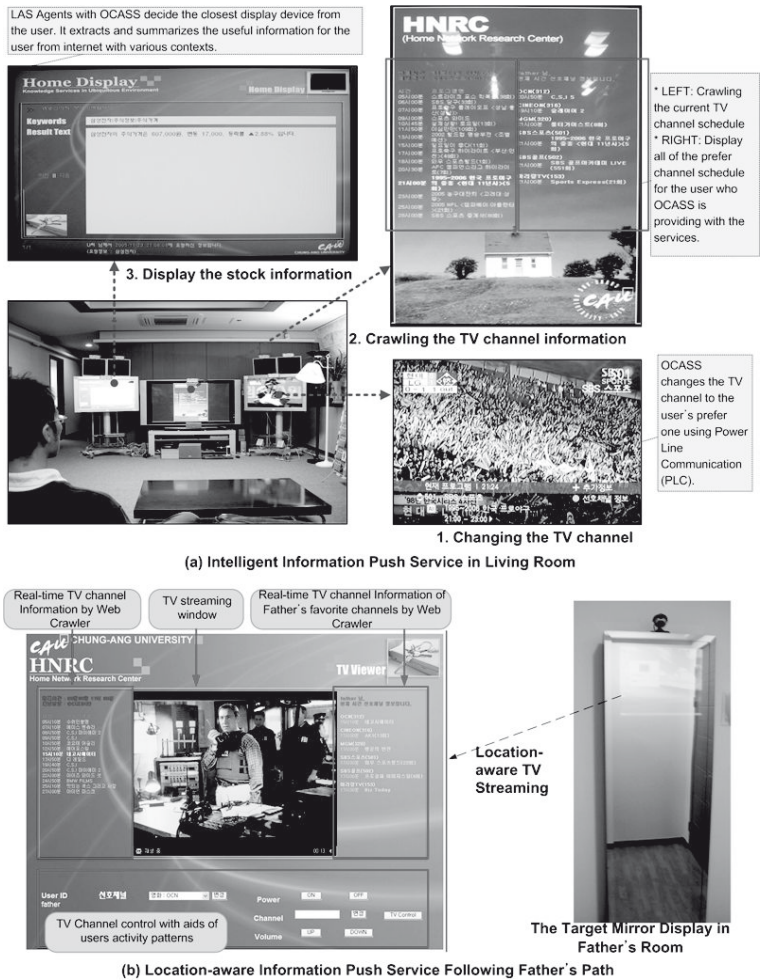


Fig. 3. Deployment of Location-aware Information Push Service in Our Smart Home

* *Intelligent Information Push Service* We trained our system recognizing and learning user's formal/informal activity patterns on *Knowledge Repository* in several weeks. The father enters the living room at 7 PM and sits on the sofa. OCASS reasons the father is at home after his work. Then OCASS evaluates the dynamic context to provide the father with more adaptive and optimal service environments as shown in Fig. 3 (a). Next, OCASS turn on the TV on the display 1 and tunes the preferred channel with the program guide of the channel on display 2. OCASS also performs the web crawling and show the stock prices of his interests on the display 3.

* *Location-aware Information Push Service with LAS Agents* The father can utilize the *Seamless Intelligent Information Push Service* provided by the most suitable LAS agents according to his location as shown in Fig. 3 (b) and Fig. 4. When the father walks through the corridor and enters his room, location sensors detect the movement of the user(Step ①, ②). Then, LAS agents are notified by the location sensors and change the context of the user following his roaming path (Step ③, ④). Next, LAS agents inform OCASS of the target spot (Step ⑤, ⑥). Finally, OCASS modifies the target devices (Step ⑦, ⑧), acquires

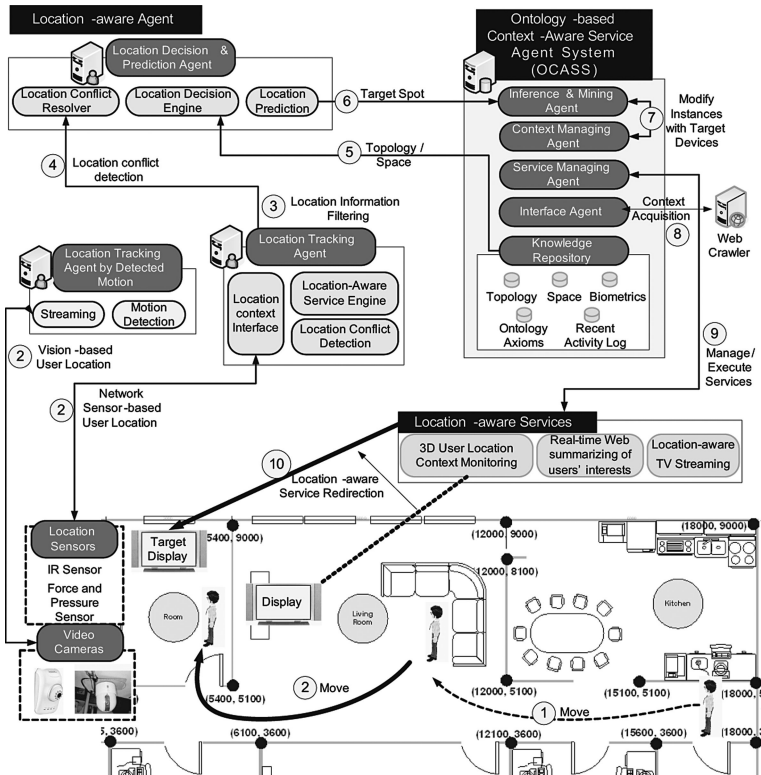


Fig. 4. Our Location-aware Service Flow with Multiple Location Sensors

context from web crawler and makes the location-aware services to be forwarded to the nearest display component in his room (Step ⑨, ⑩).

3.3 System Performance and Lessons Learned

We evaluated our model using two scenarios as shown in Table 1. The average delay of location-unaware information push service was 4,653 ms. The average delay of our user location decision scheme with LAS agents was 1,672 ms. Our location-aware information push service with LAS agents shows about the same performances and the average delay was 5,173 s by avoiding the additional delay of 3,501 ms introduced by location-aware pre-caching Web Crawling with LAS agents along the roaming path of the user. We identified the following advantage of LAS agents with our OCASS. LAS agents provide more intelligence with lower network traffic. If the number of location information requests during a service is high, so many data transmission to the *Inference & Mining Agent* can be significantly avoided by using LAS agents. By the cooperation of LAS agents and *Inference & Mining Agent* becomes more able to dynamically localize the service

Table 1. The Measured Performance of our Testbed

Entity	Operation in scenario	Performance
	Information Push Service (without Location-awareness)	
PDA ↔ AP	802.11b scan & association	40~300ms
Context Managing Agent	Modifying OWL individuals for synchronization	31ms
Inference Agent	Rule Matching & Infer Service	266ms
Service Managing Agent	Generate & Manage Service	21ms
Interface Agent ↔ e-mail Server	e-mail Push Service	219~302ms
Web Crawler	Real-time Web Information Summarizing Service	3,735ms
Service Managing Agent ↔ Streaming Server	Real-time TV Streaming Service	2,500~3,000ms
	Location-aware Information Push Service	
IR Sensor	IR-based Distance Measuring	50~80ms
Force and Pressure Sensor	Pressure Sensor-based Location Measuring	50~60ms
Network Camera ↔ Vision-based Location Tracking Agent	User Motion Detection	1,500~4,500ms
Location Tracking Agent ↔ Location Decision Agent	Location Tracking & Decision	235ms
PDA ↔ AP	802.11b scan & association	40~300ms
Context Managing Agent	Modifying OWL individuals for synchronization	31ms
Inference Agent	Rule Matching & Infer Service	266ms
Service Managing Agent	Generate & Manage Service based on Location-awareness	41ms
Interface Agent ↔ e-mail Server	e-mail Push Service	219~302ms
Mining Agent ↔ Web Crawler	Real-time Web Information Summarizing Service	234ms
Service Managing Agent ↔ Streaming Server	Real-time TV Streaming Service	2,500~3,000ms

areas of interest and adapt to change the service in shorter response time. User can utilize services and information provided by the most suitable LAS agents according to users location. If the user moves from one area to another, the actual service agent is updated dynamically without user intervention. In a similar fashion to [26] we share the participants' general feedback on our smart home with LAS agents and OCASS. The results suggest that LAS agent with OCASS deployments can be an effective tool in helping home network members. The intelligent services with LAS agents was well received both by the home network members and the elders. Participants thought that the display was pleasing and blended in nicely with their home appliances. They tended to receive the services in often used, common areas of their homes.

4 Conclusion and Future Work

In this paper, we describe an Ontology based Context-Aware Service Agent System architecture for providing context-aware services in a smart home. We discussed how OCASS system affected the quality of lives of inhabitants and their home network service members. We also shared many of our lessons learned, including successes and areas for improvement. We hope that our system helps build a body of knowledge about the use of intelligent services in the home environment. We believe that our system architecture, with the context model, should support tasks including recognizing user's informal activity pattern, learning context patterns, and resolving conflict problems between services. We will continue to develop our system to be able to interaction other agents [6][16] and systems [9][10][11] in an extended smart space. This paper leaves open the issue of how to handle the privacy controls and how to examine what happens to the acceptance of services with OCASS when a home is filled with more sensors collecting more detailed and accurate human data. Achieving intelligent services will be accomplished through new technologies and architecture utilizing the overall autonomic behaviors. Our research indicates that the autonomic cooperation of agents can be a key solution for a large and growing community of users who have a critical need for help in home network services.

References

1. Lee, M., Park, S.: A secure context management for QoS-Aware vertical handovers in 4g networks. In: Communications and Multimedia Security (CMS 2005). Volume 3677 of Lecture Notes in Computer Science., Springer (2005) 220–229
2. Hong, J.I., Landay, J.A.: An infrastructure approach to context-aware computing. *Human-Computer Interaction* **16** (2001) 287–303
3. H, C., T, F.: An ontology for a context aware pervasive computing environment. In: Eighteenth International Joint Conference On Artificial Intelligence (IJCAI) Workshop on Ontologies and Distributed Systems. (2003)
4. Tao Gu, H.K.P., Zhang, D.Q.: A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications* **28** (2005) 1–18

5. Lee, M., Park, S., Song, O.: Elastic security QoS provisioning for telematics applications. In: Workshop on Information Security Applications (WISA 2005). Volume 3786 of Lecture Notes in Computer Science., Springer (2005) 126–137
6. Lee, M., Kim, G., Park, S.: Seamless and secure mobility management with location aware service (LAS) broker for future mobile interworking networks. *Journal of Communications and Networks* **7** (2005) 207–221
7. Hazas, M., S.J.K.J.: Location-aware computing comes of age. *IEEE Computer* **37** (2004) 95–97
8. Laibowitz, M., Paradiso, J.: Parasitic mobility for pervasive sensor networks. In: Pervasive 2005. Volume 3468 of Lecture Notes in Computer Science., Springer (2005) 255–278
9. Lee, M., Kim, G., Park, S., Jun, S., Nah, J., Song, O.: Efficient 3G/WLAN interworking techniques for seamless roaming services with location-aware authentication. In: NETWORKING 2005. Volume 3462 of Lecture Notes in Computer Science., Springer (2005) 370–381
10. Lee, M., Kim, J., Park, S., Song, O., Jun, S.: A location-aware secure interworking architecture between 3GPP and WLAN systems. In: ICISC 2005. Volume 3506 of Lecture Notes in Computer Science., Springer (2004) 394–406
11. Kim, Y., Hwang, E., Uhm, Y., Park, S., Song, O.: A context aware multiagent system architecture for a smart home. In: IFIP International conference on Network Control and Engineering for QoS, Security and Mobility (NetCon05). (2005)
12. Biegel, G., Cahill, V.: A framework for developing mobile, context-aware applications. In: PerCom. (2004) 361–365
13. Lee, J., et al.: Feature fusion-based multiple people tracking. In: PCM 2005. Volume 3767 of Lecture Notes in Computer Science., Springer (2005) 843–853
14. Kurahashi, M., et al.: System support for distributed augmented reality in ubiquitous computing environments. In: RTCSA. (2003) 279–295
15. Fine, S., Singer, Y., Tishby, N.: The hierarchical hidden markov model: Analysis and applications. *Machine Learning* **32** (1998) 41–62
16. Lee, M., Kim, J., Park, S., Lee, J., Lee, S.: A secure web services for location based services in wireless networks. In: NETWORKING 2004. Volume 3042 of Lecture Notes in Computer Science., Springer (2004) 332–344
17. de Ipiña, D.L., Lo, S.L.: Locale: A location-aware lifecycle environment for ubiquitous computing. In: ICOIN. (2001) 419–426
18. Hightower, J., Borriello, G.: Particle filters for location estimation in ubiquitous computing: A case study. In: Ubicomp 2004. Volume 3205 of Lecture Notes in Computer Science., Springer (2004) 88–106
19. Hightower, J., et al.: Learning and recognizing the places we go. In: Ubicomp 2005. Lecture Notes in Computer Science, Springer (2005) 159–176
20. Liu, T., et al.: Mobility modeling, location tracking and trajectory prediction in wireless atm networks. *IEEE Selected Areas in Communications* **16** (1998) 922–936
21. Chellappa-Doss, R., et al.: User mobility prediction in hybrid and ad hoc wireless networks. In: Australian Telecommunications, Networks and Applications Conference (ATNAC). (2003)
22. The RuleML Initiative: (<http://www.ruleml.org>)
23. JESS, the Rule Engine for Java Platform: (<http://herzberg.ca.sandia.gov/jess/>)
24. Jena2, A Semantic Web Framework: (<http://www.hpl.hp.com/semweb/jena2.htm>)
25. Multimodal Interaction Framework: (<http://www.w3.org/tr/mmi-framework>)
26. Consolvo, S., Roessler, P., Shelton, B.E.: The carenet display: Lessons learned from an in home evaluation of an ambient display. In: Ubicomp 2004. Volume 3205 of Lecture Notes in Computer Science., Springer (2004) 1–17

A Verifiable Logic-Based Agent Architecture

Marco Alberti¹, Federico Chesani², Marco Gavanelli¹,
Evelina Lamma¹, and Paola Mello²

¹ ENDIF, University of Ferrara, Italy

{marco.alberti, marco.gavanelli, evelina.lamma}@unife.it

² DEIS, University of Bologna, Italy

{fchesani, pmello}@deis.unibo.it

Abstract. In this paper, we present the SCIFF platform for multi-agent systems.

The platform is based on Abductive Logic Programming, with a uniform language for specifying agent policies and interaction protocols. A significant advantage of the computational logic foundation of the SCIFF framework is that the declarative specifications of agent policies and interaction protocols can be used directly, at runtime, as the programs for the agent instances and for the verification of compliance.

We also provide a definition of conformance of an agent policy to an interaction protocol (i.e., a property that guarantees that an agent will comply to a given protocol) and a operational procedure to test conformance.

1 Introduction

In the last few years, the multi agent paradigm has been proposed as an effective conceptual and technological tool for enhancing and automating interaction between humans in areas such as workflow management and electronic commerce.

However, in order for the users to be willing to delegate critical functions to software agents, it is necessary for the agent technology to be highly reliable.

Of course, users' trust will be much higher if the reliability of Multi-agent Systems (also MAS in the following) is not only demonstrated by running systems, but also proved formally, by expressing and proving formal properties of a MAS.

The properties of a MAS that can be expressed and proved obviously depend on the amount of information that is available about the MAS and the agents that compose it.

Guerin and Pitt [1] distinguish three possible types of verification of compliance to an interaction protocol (i.e., the rules that agent should follow in their interaction), depending on the available information: *Type 1* (verify that an agent will always comply), *Type 2* (verify compliance by observation), and *Type 3* (verify protocol properties that are independent of the agents).

Type 1 verification can be performed at design time. Given a representation of the agent, by means of some proof technique (such as *model checking* [2]) it proves that the agent will always exhibit the desired behaviour.

Type 2 verification can be performed at runtime. It checks that the *actual* agent behaviour being observed is compliant to some specification. It does not require any knowledge about the agent internals, but only the observability of the agent behaviour.

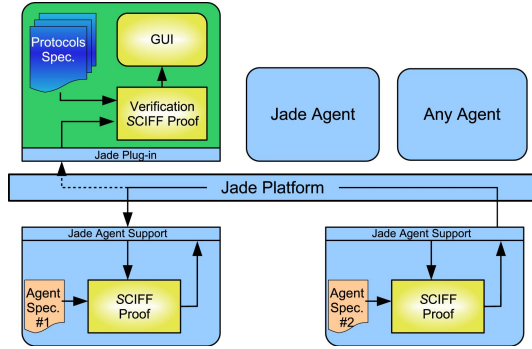


Fig. 1. The SCIFF agent platform architecture

Type 3 verification can be performed at design time. It proves that some property will hold in the society, provided that the agents follow the interaction protocols (i.e., behave accordingly to the interaction specification).

In previous work, we focused on *Type 2* and *3*: for *Type 2* verification, we developed the SCIFF abductive framework and proof procedure [3], integrated into the SOCS-SI system [4]; for *Type 3* verification, we developed the g-SCIFF extension of SCIFF [5].

In this paper, we present the SCIFF agent platform, and we tackle *Type 1* verification. For this type of verification, a representation of the agent internal policy is necessary. We developed an agent model, inspired by that proposed by Kowalski and Sadri [6] where the agent internal policies are expressed by means of the same formalism (the SCIFF language) that we use for specifying interaction protocols. This choice, besides letting us exploit the SCIFF operational machinery to implement agent systems, as we show in this paper, also makes it simpler to express and verify that an agent will be compliant to an interaction protocol.

The paper is structured as follows. In Sect. 2, we introduce the SCIFF agent platform architecture, and the language that can be used for specifying agent policies and interaction protocols. In Sect. 3, we give a definition of conformance, and propose an operational procedure to verify conformance. Related work and conclusions follow.

2 The SCIFF Agent Platform

The SCIFF agent platform, represented in Fig. 1, has been implemented on top of the JADE agent platform [7]. All the components have been implemented as JADE agents, thus exploiting the capabilities of this powerful agent platform. The single components can be used also separately, and in conjunction with other JADE-compliant agents.

Up to now, two key components have been realized: the *SOCS-SI*¹ tool, which can be used for the on-line verification, and the SCIFF agent, whose instances will populate

¹ The name stands for SOCS Social Infrastructure. SOCS is the acronym of the European project Societies of Computees (IST-2001-32530) which originally supported the research for the SCIFF framework.

the MAS. Optionally, any other agent that is able to correctly interact with the JADE platform, can take part to the interactions between these components.

The *SOCS-SI* tool, whose purpose is to observe the agent interactions and to check that they happen according to the interaction protocols, has been discussed in previous publications [4] and, for lack of space, we do not describe it here. We will stress only that *SOCS-SI* is the fundamental component that allows us to tackle *Type 2* verification, as defined by [1]. In the following we will briefly introduce also the *SCIFF* abductive framework, which we use to specify both agent policies and interaction protocols.

The *SCIFF* agent is a generic component whose behaviour is determined by its specification, provided by means of the *SCIFF* language, explained later. The agent communication language used by the *SCIFF* agents is almost the same of the one used in JADE (and defined by FIPA [8]). An important difference however is that the *SCIFF* agent is not restricted to use the pre-defined set of FIPA message performatives: the agent designer can use any desired performative. However, the semantics of such performatives should be always expressed by providing, besides the agent specification, a proper protocol, thus providing a social semantics, and possibly by means of the *SCIFF* language. Of course, careful considerations must be taken when introducing such new performatives, since inter-operability with other agent implementations is not guaranteed anymore, unless it is known a-priori that they support non-FIPA performatives.

2.1 The *SCIFF* Language

We specify the agent policies, as well as interaction protocols, by means of an abductive logic program (ALP, for short, in the following) [9].

Abductive Logic Programs. An ALP is a triple $\langle P, A, IC \rangle$, where P is a logic program, A is a set of distinguished predicates named *abducibles*, and IC is a set of integrity constraints. Reasoning in abductive logic programming is usually goal-directed (being G a goal), and corresponds to find a set of abduced hypotheses Δ built from predicates in A such that $P \cup \Delta \models G$ and $P \cup \Delta \models IC$. Suitable proof procedures (e.g., Kakas-Mancarella [10], IFF [11], SLDNFA [12], etc.) have been proposed to compute such set Δ , in accordance with the chosen declarative semantics.

Events. *Events* are our abstraction to represent the agent behaviour.

We consider two types of events: *happened* events (denoted by the functor **H**) and *expected* events (denoted by **E**, and also called *expectations*). Both are abducible, and represent hypotheses on, respectively, which events have happened and which are expected to happen, or not to happen: $\mathbf{H}(m_x(\dots), T_x)$, expresses the fact that a message m_x has been exchanged between two agents at time T_x , and $\mathbf{E}(m_x(\dots), T_x)$ says that the message m_x is expected to be exchanged at time T_x .

Protocol Specification. An interaction protocol specifies the allowed interactions among agents from an external viewpoint, i.e., it specifies their desirable observable behaviour.

We specify an interaction by means of an abductive logic program. The specification, besides representing a declarative representation of the protocol, is also used directly in

Specification 2.1. Specification of the *query_ref* interaction protocol.

Integrity constraints:

$$\begin{aligned}
& \mathbf{H}(m_x(A, B, \text{query\_ref}(\text{Info})), T) \wedge \text{qr\_deadline}(TD) \\
\rightarrow & \mathbf{E}(m_x(B, A, \text{inform}(\text{Info}, \text{Answer})), T1) \wedge T1 < T + TD \\
& \vee \mathbf{E}(m_x(B, A, \text{refuse}(\text{Info})), T1) \wedge T1 < T + TD \\
\\
& \mathbf{H}(m_x(A, B, \text{inform}(\text{Info}, \text{Answer})), Ti) \\
& \wedge \mathbf{H}(m_x(A, B, \text{refuse}(\text{Info})), Tr) \\
\rightarrow & \text{false}
\end{aligned}$$

Knowledge Base:

$$\text{qr\_deadline}(10).$$

the *SOCS-SI* tool as the program for the verification, at run time, that the agents actually comply to the protocol.

A protocol specification $\mathcal{P}_{prot}$ is defined by a tuple: $\mathcal{P}_{prot} \equiv \langle KB_{prot}, \mathcal{E}_{prot}, \mathcal{IC}_{prot} \rangle$, where:

- KB_{prot} is the *Knowledge Base*,
- $\mathcal{E}_{prot}$ is the set of *abducible predicates*, and
- $\mathcal{IC}_{prot}$ is the set of *Integrity Constraints*.

KB_{prot} specifies declaratively knowledge about the interaction protocol, such as role descriptions and the list of participants. It is a logic program whose clauses may also contain in their body expectations about the behaviour of participants, and CLP [13] constraints on variables.

The *abducible predicates* are those that can be hypothesized in our framework, namely happened events (**H**) and expectations (**E**).

Integrity Constraints are forward rules, of the form *body* $\rightarrow$ *head*, whose *body* can contain literals and (happened and expected) events, and whose *head* can either be *false*, or contain (disjunctions of) conjunctions of expectations. The syntax of $\mathcal{IC}_{prot}$ is the same defined for the SOCS Integrity Constraints [14]. CLP constraints can be imposed on variables.

In order to support goal directed interactions, we let the user specify a *goal*, which has the same syntax as the body of the clauses in KB_{prot} .

Spec. 2.1 shows the specification of the FIPA *query_ref* protocol [15].

Intuitively, the first IC means that if agent *A* sends to agent *B* a *query_ref* message, then *B* is expected to reply with either an *inform* or a *refuse* message by *TD* time units later, where *TD* is defined in the Social Knowledge Base by the *qr_deadline* predicate (with the example in Spec. 2.1, the value of *TD* would be 10).

The second IC prohibits an agent to send both an *inform* message and a *refuse* message.

Specification 2.2. A simple train timetable agent.

$$\begin{aligned} & \mathbf{H}(m_x(A, me, query\_ref(trainTime(TrainCode))), T) \wedge timeTable(TrainCode, TTime) \\ \rightarrow & \mathbf{E}(m_x(me, A, inform(trainTime(TrainCode), TTime)), TI). \end{aligned}$$

2.2 The SCIFF Agent

The same language used for the specification of the interaction protocols can be used for the specification of agent policies. In fact, it can also be used directly as the implementation language of reactive agents, following the Kowalski-Sadri schema [6].

Agent Specification. An Agent Specification $\mathcal{P}_{ag}$ is an ALP $\mathcal{P}_{ag} \equiv \langle KB_{ag}, \mathcal{E}_{ag}, \mathcal{IC}_{ag} \rangle$, where:

- KB_{ag} is the *Knowledge Base* of the agent,
- $\mathcal{E}_{ag}$ is the set of *abducible predicates*, and
- $\mathcal{IC}_{ag}$ is the set of *Integrity Constraints* of the agent.

KB_{ag} and $\mathcal{IC}_{ag}$ are completely analogous to their counterparts in the protocol specification, except that they represent an individual, rather than global, perspective: they represent, respectively, the declarative knowledge and the policies of the agent.

$\mathcal{E}_{ag}$ is the set of abducible predicates: as for the protocols, it contains expectations and happened events. The expectations can be divided into two significant subsets:

- expectations about messages where ag is the sender (of the form $\mathbf{E}_{ag}(m_x(ag, A, Content))$), i.e., actions that ag intends to do;
- expectations about messages uttered by other participants to ag (of the form $\mathbf{E}_{ag}(m_x(A, ag, Content))$, with $A \neq ag$), which can be intended as the messages that ag is able to understand.

In Spec. 2.2 a simple agent specification is shown: such agent, upon the request of a piece of information about a train (identified by the train code), always answers back with the time of that train. In the example, we assume that the `timeTable/2` predicate is defined in the knowledge base of the agent; for lack of space, we have omitted the knowledge base. We also assume that the keyword `me` is used in the expectations to identify actions that must be executed by the agent.

Architecture. The SCIFF agent has been modeled on the basis of the Kowalsky-Sadri cycle for intelligent agents [6]. In particular, the phases of *observe* and *act* have been implemented directly in Java, by extending a JADE agent. The *think* phase instead has been realized using the SCIFF proof procedure [3], that provides instructions on the basis of the happened events. A schematic representation of the blocks composing the SCIFF agent is shown in Fig. 2.2.

More in detail, the observation step consists on analyzing all the events that happened since the observation step of the previous cycle. All the events are registered in an *Event Buffer*, a repository that keeps trace of the new events.

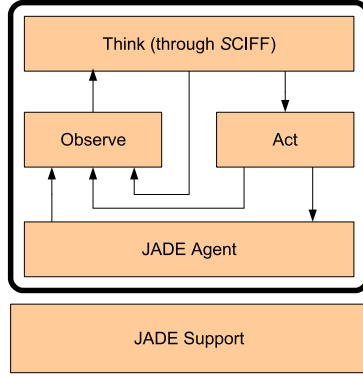


Fig. 2. Schematic diagram of the *SCIFF* Agent

The “think” step consists of using the *SCIFF* proof procedure to elaborate the happened events, and to generate a set of alternative expected behaviours. By applying a further selection function to all the alternatives, only one behaviour is selected, and it is passed on to the execution block.

The execution step consists on executing the expected behaviour generated by the *SCIFF* proof procedure, limited to the expectations about the agent’s own behaviour. For each action executed, the execution block generates a corresponding event and updates the buffer of the happened events.

3 Conformance

In this section, we define the conformance of an agent to an interaction protocol, and we propose an operational procedure to check conformance.

3.1 Definitions

The intuitive meaning of the notion of conformance is the ability of an agent to interact with other agents as specified by an interaction protocol.

Definition 1. *Given an agent specification $\mathcal{P}_{ag} \equiv \langle KB_{ag}, \mathcal{E}_{ag}, \mathcal{IC}_{ag} \rangle$ and an interaction protocol specification $\mathcal{P}_{prot} \equiv \langle KB_{prot}, \mathcal{E}_{prot}, \mathcal{IC}_{prot} \rangle$, and given the abductive program $\langle KB_U, \mathcal{E}_U, \mathcal{IC}_U \rangle$, where:*

- $KB_U \triangleq KB_{prot} \cup KB_{ag}$
- $\mathcal{E}_U \triangleq \mathcal{E}_{prot} \cup \mathcal{E}_{ag}$
- $\mathcal{IC}_U \triangleq \mathcal{IC}_{prot} \cup \mathcal{IC}_{ag}$

a possible interaction is a pair $(\mathbf{HAP}, \mathbf{EXP})$ where $\mathbf{HAP}$ is a set of events and $\mathbf{EXP}$ is a set of expectations such that

$$KB_U \cup \mathbf{HAP} \cup \mathbf{EXP} \models G \quad (1)$$

$$KB_U \cup \mathbf{HAP} \cup \mathbf{EXP} \models \mathcal{IC}_U \quad (2)$$

$$\mathbf{E}_{ag}(m_x(ag, R, M)) \rightarrow \mathbf{H}(m_x(ag, R, M)) \quad (3)$$

$$\mathbf{E}_{prot}(m_x(A, B, M)) \wedge A \neq ag \rightarrow \mathbf{H}(m_x(A, B, M)) \quad (4)$$

Conditions (1) and (2) express the requirement, usual in abductive frameworks, that the set of generated abducibles entail both the goal and the integrity constraints. Condition (3) expresses the fact that the agent will fulfill the expectations that it has about its own behaviour. Condition (4) formalizes the idea that the other agents will behave according to the protocol. These two conditions will be used as generative rules in the operational semantics. G is the goal of the derivation.

If $(\mathbf{HAP}, \mathbf{EXP})$ is a possible interaction, then $\mathbf{HAP}$ is a *possible history*.

Example 1. Given the specifications in Specs. 2.1 and 2.2, a possible interaction is given by

$$\begin{aligned} \mathbf{HAP} &= \{\mathbf{H}(m_x(b, a, \text{query\_ref}(\text{trainTime}(10))), 1), \mathbf{H}(m_x(a, b, \text{inform}(\text{trainTime}(12))), 2)\} \\ \mathbf{EXP} &= \{\mathbf{E}_{ag}(m_x(a, b, \text{inform}(\text{trainTime}(I'))), T'), \mathbf{E}_{prot}(m_x(a, b, \text{inform}(\text{trainTime}(I'))), T')\} \end{aligned}$$

We can now provide a definition of *conformance* based on the possible interactions, selecting those that indeed respect the protocol and agent specifications.

Definition 2 ($\exists$ -Conformance). *An agent specification $\mathcal{P}_{ag}$ is existentially conformant to a protocol specification $\mathcal{P}_{prot}$ if there exists at least one possible interaction $(\mathbf{HAP}, \mathbf{EXP})$ such that the following implications hold:*

$$\mathbf{E}_{ag}(m_x(A, B, M)) \rightarrow \mathbf{H}(m_x(A, B, M)) \quad (5)$$

$$\mathbf{E}_{prot}(m_x(A, B, M)) \rightarrow \mathbf{H}(m_x(A, B, M)) \quad (6)$$

Condition 5 (resp. 6) requires that if ag (resp. the protocol) had an expectation for a message to be sent, the corresponding event should have happened.

Example 2. The agent specified in Spec. 2.2 is $\exists$ -conformant to the protocol specified in Spec. 2.1, because the possible interaction in Eq. (1) also respects the conditions in Eqs. (5) and (6).

Definition 3 ($\forall$ -Conformance). *An agent specification $\mathcal{P}_{ag}$ is universally conformant to a protocol specification $\mathcal{P}_{prot}$ if for each possible history $\mathbf{HAP}$ there exists a set of expectations $\mathbf{EXP}$ such that $(\mathbf{HAP}, \mathbf{EXP})$ is a possible interaction and the conditions 5 and 6 hold for it.*

When Definitions 2 and 3 hold together for an agent ag , then ag is *conformant*.

3.2 Operational Test

The operational semantics is based on the two versions of the SCIFF proof procedure [3] developed in the SOCS project. The SCIFF proof procedure was proven sound [16]

and complete [17]; *SCI*FF terminates [16] for acyclic programs. The *SCI*FF proof procedure considers the **H** events as a predicate defined by a set of incoming atoms, and is devoted to generate expectations corresponding to a given history and to check that expectations indeed match with happened events. *SCI*FF was developed to check the compliance of agents to protocols [3]. The *SCI*FF proof procedure is based on a rewriting system transforming one node to another (or to others) as specified by rewriting steps called *transitions*.

A generative version of *SCI*FF, called g-*SCI*FF, instead, considers **H** as an abducible predicate and aims at finding both the set of expectations and the history that fulfils some property requested as goal. g-*SCI*FF has been used to prove properties of protocols, such as security protocols [18], and others. It contains the same rules in *SCI*FF; in the version adopted in this paper, we also added as integrity constraints the equations (3) and (4).

In order to prove conformance, we apply the two versions of the proof procedure to the two phases implicitly defined in the previous section. We decompose the proof of conformance into a *generative* phase and a *test* phase. In the generative phase, we generate, by means of g-*SCI*FF, all the possible histories. Of course, those histories need not be generated as ground histories (the set of ground histories can be infinite), but intensionally: the **H** events can contain variables, possibly with constraints à la Constraint Logic Programming [13].

In the test phase, we check with *SCI*FF if the generated histories are conformant. If all the histories are conformant, the agent is conformant to the protocol: at runtime, all the possible agent instances will behave accordingly to the protocol. Otherwise, if there exists at least one history that is not conformant, the agent is not conformant.

4 Related Work

Our work is highly inspired by Baldoni et al. [19], who adopt a Multi-agent Systems point of view in defining a priori conformance in order to guarantee interoperability of Web Services whose interactions are specified by choreographies. As in [19], we give an interpretation of the a-priori conformance as a property that relates two formal specifications: the global one determining the interactions allowed by the interaction protocols and the local one related to the single agent policies. But, while in [19] a specification is represented as a finite state automation, we claim that the formalisms and technologies developed in the area of Computational Logic in providing a declarative representation of the social knowledge, could be applied also in the context of interaction protocol with respect to the conformance checking of agents. For example, a difference between our work and [19] can be found in the number of parties as they can manage only 2-party choreographies while we do not impose any such limit on the number of interacting agents.

In [20,21], the authors apply a formalism based on computational logic to the a-priori conformance in the MAS field. Their formalism is similar to the one we propose, but they restrict their analysis to a particular type of protocols (named *shallow protocols*). Doing this, they address only 2-party interactions, without the possibility of expressing conditions over the content of the exchanged messages.

The use of abduction for verification was also explored in other work. Noteworthy, Russo et al. [22] use an abductive proof procedure for analysing event-based requirements specifications. Their method uses abduction for analysing the correctness of specifications, while our system is more focussed on the check of compliance/conformance of a set of agents.

5 Conclusions

In this paper, we introduced an agent platform based on the *SCIFF* abductive framework. The main features of the framework are the possibility to define and check the conformance of an agent to a protocol, and the fact that the specification of an agent can be used, directly, as its implementation as a reactive system.

We described the architecture of the system, the language for the specification of the agent policies and the interaction protocols, a declarative definition of conformance of an agent to a protocol, and an operational procedure to verify conformance.

Acknowledgements

This work has been partially supported by the MIUR PRIN 2005 projects *Specification and verification of agent interaction protocols* and *Vincoli e preferenze come formalismo unificante per l'analisi di sistemi informatici e la soluzione di problemi reali*.

References

1. Guerin, F., Pitt, J.: Proving properties of open agent systems. In Castelfranchi, C., Lewis Johnson, W., eds.: *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002), Part II*, Bologna, Italy, ACM Press (2002) 557–558
2. Merz, S.: Model checking: A tutorial overview. In Cassez, F., Jard, C., Rozoy, B., M.D.Ryan, eds.: *Modeling and Verification of Parallel Processes*. Number 2067 in *Lecture Notes in Computer Science*, Springer-Verlag (2001) 3–38
3. Alberti, M., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: The *SCIFF* abductive proof-procedure. In: *Proceedings of the 9th National Congress on Artificial Intelligence, AI*IA 2005*. Volume 3673 of *Lecture Notes in Artificial Intelligence*., Springer-Verlag (2005) 135–147
4. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Compliance verification of agent interaction: a logic-based tool. *Applied Artificial Intelligence* **20** (2006) 133–157
5. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Security protocols verification in abductive logic programming: a case study. In Dikenelli, O., Gleizes, M.P., Ricci, A., eds.: *ESAW 2005 Post-proceedings*. Number 3963 in *LNAI*, Kusadasi, Aydin, Turkey, Springer-Verlag (2006) 106–124
6. Kowalski, R.A., Sadri, F.: From logic programming towards multi-agent systems. *Annals of Mathematics and Artificial Intelligence* **25** (1999) 391–419

7. Bellifemine, F., Bergenti, F., Caire, G., Poggi, A.: Jade - a java agent development framework. In Bordini, R.H., Dastani, M., Dix, J., Fallah-Seghrouchni, A.E., eds.: *Multi-Agent Programming: Languages, Platforms and Applications*. Volume 15 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*. Springer-Verlag (2005) 125–147
8. (FIPA: Foundation for Intelligent Physical Agents) Home Page: <http://www.fipa.org/>.
9. Kakas, A.C., Kowalski, R.A., Toni, F.: Abductive Logic Programming. *Journal of Logic and Computation* **2** (1993) 719–770
10. Kakas, A.C., Mancarella, P.: On the relation between Truth Maintenance and Abduction. In Fukumura, T., ed.: *Proceedings of the 1st Pacific Rim International Conference on Artificial Intelligence, PRICAI-90, Nagoya, Japan, Ohmsha Ltd.* (1990) 438–443
11. Fung, T.H., Kowalski, R.A.: The IFF proof procedure for abductive logic programming. *Journal of Logic Programming* **33** (1997) 151–165
12. Denecker, M., Schreye, D.D.: SLDNFA: an abductive procedure for abductive logic programs. *Journal of Logic Programming* **34** (1998) 111–167
13. Jaffar, J., Maher, M.: Constraint logic programming: a survey. *Journal of Logic Programming* **19-20** (1994) 503–582
14. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: The SOCS computational logic approach for the specification and verification of agent societies. In Priami, C., Quaglia, P., eds.: *Global Computing: IST/FET International Workshop, GC 2004 Rovereto, Italy, March 9-12, 2004 Revised Selected Papers*. Volume 3267 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag (2005) 324–339
15. FIPA Query Interaction Protocol (2001) Published on August 10th, 2001, available for download from the FIPA website, <http://www.fipa.org>.
16. Gavanelli, M., Lamma, E., Mello, P.: Proof of properties of the SCIFF proof-procedure. (Technical Report CS-2005-01)
17. Gavanelli, M., Lamma, E., Mello, P.: Proof of completeness of the SCIFF proof-procedure. (Technical Report CS-2005-02)
18. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Security protocols verification in Abductive Logic Programming: A case study. In Pettorossi, A., Proietti, M., Senni, V., eds.: *CILC 2005 - Convegno Italiano di Logica Computazionale*, Università degli Studi di Roma Tor Vergata (2005)
19. Baldoni, M., Baroglio, C., Martelli, A., Patti, V., Schifanella, C.: Verifying the conformance of web services to global interaction protocols: A first step. In Bravetti, M., Kloul, L., Zavattaro, G., eds.: *EPEW/WS-FM*. Volume 3670 of *Lecture Notes in Computer Science*., Springer (2005)
20. Endriss, U., Maudet, N., Sadri, F., Toni, F.: Protocol conformance for logic-based agents. In Gottlob, G., Walsh, T., eds.: *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico (IJCAI-03)*, Morgan Kaufmann Publishers (2003)
21. Endriss, U., Maudet, N., Sadri, F., Toni, F.: Logic-based agent communication protocols. In Dignum, F., ed.: *Advances in Agent Communication*. Volume 2922 of *LNAI*. Springer-Verlag (2004) 91–107
22. Russo, A., Miller, R., Nuseibeh, B., Kramer, J.: An abductive approach for analysing event-based requirements specifications. In Stuckey, P., ed.: *Logic Programming, 18th International Conference, ICLP 2002*. Volume 2401 of *Lecture Notes in Computer Science*., Berlin Heidelberg, Springer-Verlag (2002) 22–37

Flexible Querying of XML Documents

Krishnaprasad Thirunarayan and Trivikram Immaneni

Department of Computer Science and Engineering
Wright State University, Dayton, OH 45435, USA
t.k.prasad@wright.edu, immaneni.2@wright.edu

Abstract. Text search engines are inadequate for indexing and searching XML documents because they ignore metadata and aggregation structure implicit in the XML documents. On the other hand, the query languages supported by specialized XML search engines are very complex. In this paper, we present a simple yet flexible query language, and develop its semantics to enable intuitively appealing *extraction* of relevant fragments of information while simultaneously falling back on *retrieval* through plain text search if necessary. We also present a simple yet robust relevance ranking for heterogeneous document-centric XML.

1 Introduction and Motivation

Popular search engines (such as Google, Yahoo!, MSN Search, etc) index document text and retrieve documents efficiently in response to easy to write queries. Unfortunately, these search engines suffer from at least two drawbacks: (a) They ignore information available in the metadata / annotations / XML tags¹. (b) They are oblivious to the underlying aggregation structure implicit in the tree-view of XML documents. On the other hand, specialized search engines for querying XML documents require some sophistication on the part of the user to formulate queries.

In this paper, we present a query language that is simple to use and yields results that are more meaningful than the corresponding text search would yield on an XML document. Specifically, it facilitates exploitation of metadata to *extract* relevant fragments of information without sacrificing the ability to fall back on *retrieval* through plain text search.

Example 1. Consider the following fragment² of SIGMOD record [2].

```
- <article>
  <title>A Note on Decompositions of Relational Databases.</title>
- <authors>
  <author position="00">Catriel Beeri</author>
  <author position="01">Moshe Y. Vardi</author>      </authors>    </article>
```

¹ Actually, search engines do analyze the content associated with the META-element, the TITLE-element, etc, and factor in information implicit in the text fonts and anchor text (link analysis), for relevance ranking an HTML document [1].

² The word "Database" in the second record appears as "Data Base" in the original dataset.

```

- <article>
  <title>Implementation of a Time Expert in a Data Base System.</title>
- <authors>
  <author position="00">Ricky Overmyer</author>
  <author position="01">Michael Stonebraker</author>    </authors> </article>

```

Searching for “articles by Stonebraker” should retrieve the above page, and better yet, cull out information about the title, co-authors, etc, of Stonebraker’s articles while ignoring articles by others. Similarly, searching for “articles with database” in their title should ideally yield all the above articles.

Example 2. Consider the following fragment of Mondial database [2]. Observe that a lot of factual data is captured via attribute bindings.

```

<mondial>
<continent id="f0_119" name="Europe" /> ...
  <country id="f0_149" name="Austria" capital="f0_1467" population="8023244"
    datacode="AU" total_area="83850" population_growth="0.41"
    infant_mortality="6.2" ... government="federal republic" ...>
    <name>Austria</name> ...
  - <province id="f0_17447" name="Vienna" ...>
    - <city id="f0_1467" country="f0_149" province="f0_17447" ...>
      <name>Vienna</name>    <population year="94">1583000</population>    </city>
    </province> ...
  <languages percentage="100">German</languages>
  <encompassed continent="f0_119" percentage="100" /> ...
  <border length="784" country="f0_220" /> ...
</country> ...
</mondial>

```

Searching for “Austria” should fetch the above record from the database from which further geographic information about its cities and provinces can be determined. The internal name/code for Austria in the database can be used to infer additional information related to bordering countries, border lengths, etc.

Example 3. Consider the following heterogeneous XML fragment [4].

```

<document>
<article id="1">
  <author><name>Adam Dingle</name></author>
  <author><name>Peter Sturmh</name></author>
  <author><name>Li Zhang</name></author>
  <title>Analysis and Characterization of Large-Scale Web Server Access
    Patterns and Performance</title>
  <year>1999</year>
  <booktitle>World Wide Web Journal</booktitle>    </article>
<article id="2" year="1999">
  <author name="A. Dingle" ></author>
  <author name="E. Levy" ></author>
  <author name="J. Song" ></author>
  <author name="D. Dias" ></author>
  <title>Design and Performance of a Web Server Accelerator</title>
  <booktitle> Proceedings of IEEE INFOCOM </booktitle>    </article>
<article id="3">
  @inproceedings{IMN97,
    author="Adam Dingle and Ed MacNair and Thao Nguyen",
    title="An Analysis of Web Server Performance",
    booktitle="Proceedings of the IEEE Global Telecommunications
      Conference (GLOBECOM)",
    year=1999}    </article>
</document>

```

It contains information about articles expressed in three different ways. A robust search strategy should ideally deliver all the three records when articles by “Dingle” are sought.

Example 4. Similar issues arise in the context of XML representation of binary relationships and XML serialization of RDF model, as illustrated below [15]. Specifically, “same” information may be expressed differently based on the preferences and views of the authors of web documents.

```
<course name="Discrete Mathematics">
  <lecturer>David Billington</lecturer> </course>
<lecturer name="David Billington">
  <teaches>Discrete Mathematics</teaches> </lecturer>
<teachingOffering>
  <lecturer>David Billington</lecturer>
  <course>Discrete Mathematics</course> </teachingOffering>
```

Our work builds on and extends the seminal work of Cohen et al [3] on XSEarch, a search engine for XML documents. XSEarch’s query language embodies the simplicity of Google-like search interface (easing the task of query formulation) while exploiting the hierarchical structure of nested XML-elements to deliver precise and coherent results (that is, containing semantically related pieces of information). However, their query language ignores XML-attributes entirely. In this paper, we smoothly extend XSEarch’s query language to accommodate XML-attributes and their string values with the following benefits:

- XML documents with attributes can now be queried. Observe that, in the document centric applications of XML, information bearing strings are in text nodes, while in the data centric applications of XML, information bearing strings are associated with attributes.
- In spite of having general rules of thumb about when to use XML-elements and when to use XML-attributes for expressing a piece of information [14], it is still quite common to see authoring variations that mix the two. For instance, one can find the following “equivalent” pattern in use: `<T A="s"/>` and `<T> <A> s </A> </T>`. Accommodating this variation can improve query *recall*.
- Semantic Web formalisms such as RDF, OWL, etc [16,15,17] build on XML and make extensive use of attributes. In fact, the above two forms are equivalent in RDF. So a simple XML Search Engine that can deal with attributes will be a welcome addition to the toolset till customized search engines for RDF and OWL become commonplace.

Furthermore, in many applications, XML documents may get progressively refined via a sequence of annotaters. For instance, in an initial step, an entire name in a text may be recognized and enclosed within `<name> ... </name>` tags, while in a subsequent step, it may be refined by delimiting the first name and the last name using `<firstName> ... </firstName>` and `<lastName> ... </lastName>` tags respectively.

In Section 2, we present selected related works. In Section 3, we discuss the proposed XML query language, develop its semantics, and illustrate it through

examples. In Section 4, we conclude with suggestions for future work. (Figures, other related works and implementation details have been skipped due to space limitations.)

2 Selected Related Work

Florescu et al [4] present an extension of the XML-QL language that supports querying of XML documents based on its structure and its textual content, facilitating search of XML documents whose structure is only partially known, or searching heterogeneous XML document collections. The implementation uses an off-the-shelf relational database system. In XIRCL, Fuhr and Grojohann [5] incorporate different answer granularity, robust query matching, and IR-related features such as relevance ranking using probabilistic models. Meyer et al [6] describe a search engine architecture and implementation based on XIRCL. Carmel et al [7] use XML fragments as queries instead of inventing a new XML query language, and extend the traditional vector space model for XML collections. The weight associated with an individual term depends on its context, and the ranking mechanism is used to deal with imperfect matches and high recall. In comparison to all these approaches, our query language is less expressive, the weighting mechanism and ranking is simpler, given that the query answer has already culled out the relevant answer. Schlieder and Meuss [8] reduce XML querying to tree matching by simulating attributes via elements and strings via word-labelled node sequences, and adapt traditional IR techniques for document ranking. Our approach resembles this work in spirit, but again the query language is simpler. For example, queries involving an element name and a keyword has different interpretation which is robust with respect to the level of annotations. In contrast with approaches so far, Theobald and Weikum [9] focus on heterogeneous documents, path-based queries and semantic-similarity search conditions. Grabs and Schek [10] and Guo et al [13] try to capture the intuition that the content that is more distant in a document tree is less important than the one that is close to the context node while determining term weights and relevance. On the other hand, we want to preserve the semantic impact of a piece of text irrespective of the level of annotations surrounding it. Li et al [11] propose an extension to XQuery that marries the precision of XQuery with the convenience of a keyword-based XML query language. Catania et al [12] provide a nice review of the indexing schemes employed for querying XML documents.

3 Query Language

The standard search engine query is a list of optionally signed keywords. For querying XML documents, Cohen et al [3] allow users to specify labels and keyword-label combinations that must or may appear in a satisfying XML document fragment. Our queries allow keywords, element names, and attribute names, optionally with a plus (“+”) sign. Intuitively, element/attribute names relate to type information/metadata, while keywords relate to concrete values/data.

3.1 Query Syntax

Formally, a *search term* has one of the following forms: $e:a:k$, $e:a:$, $:a:k$, $e:k$, $e::$, $:a:$, $:k$, $l:k$, $l:$ and $:k$, where e is an element name, a is an attribute name, l is an element/attribute name, and k is a keyword (string). Furthermore, $l:k$ is interpreted as $l:k$ or $:l:k$, $l:$ is interpreted as $l:$ or $:l:$, and $:k$ is interpreted as $:k$. A *query* is a sequence of optionally signed search terms. Signed search terms are *required* to be present in the retrieved results, while unsigned search terms are *desirable* in the retrieved results.

3.2 XML Datamodel

Conceptually, an XML document is modelled as an ordered tree [17]. For our purposes, XML tree contains the following types of nodes³:

- *Root node*: The root node is the root of the tree. The element node for the document element is a child of the root node.
- *Element node*: There is an element node for every element in the document. The children of an element node are the element nodes and the text nodes (for its content).
- *Text node*: Character data is grouped into text nodes.
- *Attribute node*: An element node can have an associated set of attribute nodes; the element is the parent of each of these attribute nodes; however, an attribute node is not technically a child of its parent element. Each attribute node has a string-value.

3.3 Single Search Term Satisfaction

We specify when a search term is *satisfied* by an XML subtree⁴. In contrast to Cohen's work, our approach abstracts from differences in the representation of a piece of information either as an attribute-value pair of an element or as the element's subelement enclosing the value text (for example, `<T A="s"/>` and `<T> <A> s </A> </T>`), among other things. Additionally, it addresses the situation where the text of an XML document can be further refined through annotation.

- The search term $e:a:k$ is satisfied by a tree containing a subtree with the top element e that is associated with the attribute a with value containing k , or a subelement a with descendant text node containing k .
- The search term $e:a:$ is satisfied by a tree containing a subtree with the top element e that is associated with the attribute a , or subelement a .
- The search term $:a:k$ is satisfied by a tree containing a subtree with a top element that is associated with the attribute a with value containing k , or a subelement a with descendant text node containing k .

³ We ignore *namespace* nodes, *processing instruction* nodes, and *comment* nodes, and the ability to create additional internal *links* between tree nodes.

⁴ In this paper, a subtree is always rooted at an element node.

- The search term $e::k$ is satisfied by a tree containing a subtree with the top element e and that has
 - an attribute associated with the value containing k , or
 - a descendant element with an associated attribute value containing k , or
 - a descendant text node containing k .
- The search term $e::$ is satisfied by a tree containing a subtree with the top element name e .
- The search term $:a:$ is satisfied by a tree containing a subtree with a top element that is associated with the attribute a or a subelement a .
- The search term $::k$ is satisfied by a tree containing
 - a subtree with a top element that is associated with an attribute value containing k , or
 - the descendant text node containing k .

Observe that a query can use detailed knowledge of XML document structure (for example, via $e::a:k$ etc), or have the flexibility to express textual search (for example, via $:k$, etc). For the purposes of the above definition, a tree contains itself.

Example 1 (cont.) The search term $author::$ is satisfied by

```
<author position="00">Catriel Beeri</author>
<author position="01">Moshe Y. Vardi</author>
<author position="00">Ricky Overmyer</author>
<author position="01">Michael Stonebraker</author>
```

and also by trees containing such subtrees. The most preferred tree satisfying $:Stonebraker$ is `<author position="01">Michael Stonebraker</author>`. Similarly, the most preferred trees satisfying $title::Data$ are

```
<title>A Note on Decompositions of Relational Databases.</title>
<title>Implementation of a Time Expert in a Data Base System.</title>
<title>Problems of Optimistic Concurrency Control in Distributed Database Systems.</title>
```

Example 2 (cont.) The search term $country::Austria$ is satisfied by

```
<country id="f0_149" name="Austria" capital="f0_1467" population="8023244"
  datacode="AU" total_area="83850" population_growth="0.41"
  infant_mortality="6.2" ... government="federal republic" ...>
</country>
```

The search term $:name:Vienna$ is satisfied by

```
<province id="f0_17447" name="Vienna" ...>
  <city id="f0_1467" country="f0_149" province="f0_17447" ...>
    <name>Vienna</name>    <population year="94">1583000</population>    </city>
  </province> ...
```

but $name::Vienna$ is satisfied only by a part of it, that is, `<name>Vienna</name>`. The latter is also the most preferred tree satisfying $:name:Vienna$. (Observe that this query response seems to “verify existence” as opposed to “extract answer”.)

Example 3 (cont.) All the three *articles* given in Example 3 in Section 1 satisfy the search term $article::Dingle$. Similarly, the search term $:Dingle$ satisfies the following three records, while the search term $author:Dingle$ misses the last one (as they both belong to the text content).

```

<author><name>Adam Dingle</name></author>
<author name="A. Dingle" ></author>
<article id="3">
  @inproceedings{IMN97,
    author="Adam Dingle and Ed MacNair and Thao Nguyen",
    title="An Analysis of Web Server Performance",
    booktitle="Proceedings of the IEEE Global Telecommunications
      Conference (GLOBECOM)", year=1999}
</article>

```

3.4 Query Satisfaction

In order for a collection of XML subtrees to satisfy a query, *each* required (resp. optional) search term in the query must (resp. should) be satisfied by *some* subtree in the collection. The notion of satisfaction can be formalized via subtree sequences.

Similarly to Cohen et al [3], a query $Q(t_1, t_2, \dots, t_m)$ is satisfied by a sequence of subtrees and null values $(T_1, T_2, \dots, T_m)$, if

- For all $1 \leq i \leq m$: if t_i is a signed/plus/required term, then T_i is not the null value.
- For all $1 \leq i \leq m$: if T_i is not the null value, then t_i is satisfied by T_i .

We also say that the set $\{T_i | 1 \leq i \leq m \wedge T_i \text{ is non-null}\}$ satisfies Q .

3.5 Query Answer

The definition of query answer captures precision, adequacy and coherence of extracted results. A *query answer candidate* for a query $Q(t_1, t_2, \dots, t_m)$ with respect to an XML tree T is a collection of XML subtrees $\mathcal{U}$ of T such that $\mathcal{U}$ satisfies Q , and furthermore, there does not exist another (distinct) satisfying collection of XML subtrees $\mathcal{P}$ that is preferred to $\mathcal{U}$. (Note that $\mathcal{P}$ is *preferred to* $\mathcal{P}$, according to the following definition.)

Consider the subtree sequences $(P_1, P_2, \dots, P_m)$ and $(U_1, U_2, \dots, U_m)$ such that $\mathcal{U} = \{U_i \mid 1 \leq i \leq m \wedge U_i \neq \text{null}\}$ and $\mathcal{P} = \{P_i \mid 1 \leq i \leq m \wedge P_i \neq \text{null}\}$. $\mathcal{P}$ is *preferred to* $\mathcal{U}$ if and only if

1. (Precision) $\forall i : t_i \text{ is a term} \Rightarrow (P_i = U_i) \vee (P_i \text{ is embeddable / is a subtree of } U_i)$, or
2. (Adequacy[Maximal Information])
 $\forall i : t_i \text{ is an unsigned term} \Rightarrow (P_i = U_i) \vee (P_i \neq \text{null})$.

The precision criteria captures preference for the smallest subtree that is necessary to demonstrate satisfaction. Unfortunately, for certain keyword queries (such as $::k$), this can yield an answer that seems over-specific.

The adequacy or maximal information criteria captures preference for answers that provide information related to desirable terms, in addition to mandatory information for required terms.

A *query answer* for a query $Q(t_1, t_2, \dots, t_m)$ with respect to an XML tree T is a collection of XML subtrees $\mathcal{U}$ of T such that $\mathcal{U}$ is *interconnected*. Two subtrees

T_a and T_b are said to be *interconnected* if the path from their roots to the lowest common ancestor does not contain two distinct nodes with the same element, or the only distinct nodes with the same element are these roots.

The intuition behind interconnected-ness is that if the common ancestor can be viewed as a collection containing multiple entities of the same type, as evidenced by the same node label, then interconnected nodes are related to the same “physical” entity. This can be viewed as coherence criteria.

Furthermore, we can display portions of the *query answer* that are not redundant (that is, skip embedded subtrees that may be repeated). This can be viewed as conciseness criteria.

Example 1 (cont.) The query *authors::, title::* returns three authors-title pairs, and the query *author::, title::* returns four author-title pairs.

Example 3 (cont.) The first two *articles* given in Example 3 in Section 1 match the query *author:name:Dingle, article::*.

Example 4 (cont.) The query *lecturer::Mathematics* results in:

```
<lecturer name="David Billington">
  <teaches>Discrete Mathematics</teaches> </lecturer>
```

while the query *lecturer::,Mathematics* results in three answers:

```
<course name="Discrete Mathematics">
  <lecturer>David Billington</lecturer> </course>
<lecturer name="David Billington">
  <teaches>Discrete Mathematics</teaches> </lecturer>
<lecturer>David Billington</lecturer>
<course>Discrete Mathematics</course>
```

3.6 Ranking Query Answers

In order to deal with document-centric applications of XML, we adapt the traditional TFIDF formula for weighting documents to rank order query answers, somewhat along the lines of Cohen et al [3]. Specifically, we capture the relevance of an XML subtree (containing text) to a query term (containing keyword). The term frequency of a keyword k in a tree T_n , is defined as:

$$tf(k, T_n) = \frac{count(k, T_n)}{\max\{count(i, T_n) \mid i \text{ in } words(T_n)\}},$$

where $count(k, T_n)$ is the number of times k is contained in the text/attribute nodes of the tree T_n , and $words(T_n)$ is the set of keywords contained in the text/attribute nodes of the tree T_n . Similarly, the inverse document frequency of a keyword k for a subtree with root label / type t is defined as:

$$idf(k, t) = \log \left(1 + \frac{|T(t)|}{|\{T_n \in T(t) \mid k \text{ in } words(T_n)\}|} \right),$$

where $T(t)$ is the set of tree/attribute of type t .

In the context of queries involving elements and/or attributes and keywords, the *rank* of a query answer $(T_1, T_2, \dots, T_m)$ for the query $Q(t_1, t_2, \dots, t_m)$ is

$$\sum \{ tfidf(T_i, t_i) \mid 1 \leq i \leq m \}.$$

For $t_i = t : k \vee t_i = t : \_ : k \vee t_i = \_ : t : k$, $tfidf(T_i, t_i) = tf(k, T_i) * idf(k, t)$. For $t_i = :: k$, in the absence of context, the inverse document frequency of a keyword k can be changed to:

$$idf(k) = \log \left(1 + \frac{|T|}{|\{T_n \in T \mid k \text{ in } words(T_n)\}|} \right),$$

where T is the set of text/attribute nodes.

This reduces to traditional TFIDF approach for keyword search if a collection of text documents are glued into a single XML document creating one text node per text document. Observe also that the relative rank of the modified XML subtrees does not change if the XML documents are refined by embedding new element tags. Similarly, the relative ranks may be preserved when the XML documents are augmented with attribute-value pairs that reflect the semantics of a piece of text.

To implement this language, we need to build an index, mapping words to paths in the XML document tree, to enable retrieval of subtrees. To compute interconnectedness, parent relationship is essential. Our current implementation uses Lucene [18] for efficient indexing and search of XML documents and does not deal with ranking. Relative to traditional IR, we expect the size of the XML term-document matrix to be very large. Thus, instead of materializing all the TFIDF statistics for each subtree, it may be dynamically computed using the corresponding statistics for the text/attribute nodes contained in the subtree.

4 Conclusions and Future Work

The proposed XML query language has been designed with a straightforward syntax to ease query formulation, incorporating not only text data content but also metadata information in the source XML document. The query semantics has been designed in such a way that the answers provide relevant information and are robust with respect to various manifestations of the same information in terms of XML elements and XML attributes, to improve recall. In other words, the intuitive duality between the usages of elements and attributes has been taken into account. Unfortunately, in relation to traditional IR, certain keyword queries may yield answers that are over-specific from user's perspective, necessitating inclusion of metadata information in the query to control the granularity of answers. One possible approach to overcome this problem is to specify what kinds of subtrees (in terms of root labels) should be allowed in the answers.

Acknowledgements. We thank the referees for their valuable feedback.

References

1. Brin, S. and Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine, In: Proceedings of the Seventh International Conference on World Wide Web, 1998, pp. 107 - 117.
2. <http://www.cs.washington.edu/research/xmldatasets/>, Retrieved June/2006.
3. Cohen, S., Mamou, J., Kanza, Y., and Sagiv, Y.: *XSearch: A Semantic Search Engine for XML*, In: The 29th International Conference on Very Large Databases (VLDB). September 2003.
4. Florescu, D., and Donald Kossman and Ioana Manolescu: *Integrating keyword search into XML query processing*, Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 33, Issue 1-6, June 2000, pp. 119-135.
5. Fuhr, N., and Grojohann, K.: *XIRQL: A Query Language for Information Retrieval in XML Documents*, In: Proceedings of the 24th ACM SIGIR Conference, 2001, pp. 172-180.
6. Meyer, H., Bruder, I., Weber, G. and Heuer, A.: *The Xircus Search Engine*, 2003.
7. Carmel, D., Maarek, Y.S., Mass, Y., Efraty, N., and Landau, G.M.: *An Extension of the Vector Space Model for Querying XML Documents via XML Fragment*, The 25th Annual ACM SIGIR Conference, 2002.
8. Schlieder, T. and Meuss, H.: *Querying and ranking XML documents*, Journal of the American Society for Information Science and Technology, Volume 53 , Issue 6, 2002, pp. 489-503.
9. Theobald, A., and Weikum, G.: *The Index-Based XXL Search Engine for Querying XML Data with Relevance Ranking*, 8th International Conference on Extending Database Technology (EDBT), LNCS 2287, 2002, pp. 477-495.
10. Grabs, T., and Schek, H.: *Generating Vector Spaces On-the-fly for Flexible XML Retrieval*, In: Proceedings of the 2nd XML and Information Retrieval Workshop, 25th ACM SIGIR Conference, 2002.
11. Li, Y., Yu, C. and Jagadish, H. V.: *Schema-Free XQuery*, In: The 30th International Conference on Very Large Databases (VLDB), 2004, pp. 72-83.
12. Catania, B., Maddalena, A., and Vakali, A.: *XML Document Indexes : A Classification*, In: IEEE Internet Computing, 2005, pp. 64-70.
13. Guo, L., Shao, F., Botev C., and Shanmugasundaram, J.: *XRANK: Ranked keyword search over XML documents*, In: Proceedings of ACM SIGMOD, 2003, pp. 16-27.
14. <http://xml.coverpages.org/elementsAndAttrs.html>, Retrieved June/2006.
15. Antoniou, G., and van Harmelen, F.: *A Semantic Web Primer*, The MIT Press, 2004.
16. Fensel, D., Hendler, J., Lieberman, H., and Wahlster, W. (Eds.): *Spinning the Semantic Web: Bringing the WWW to Its Full Potential*, The MIT Press, 2003.
17. <http://www.w3.org/TR/>, Retrieved June/2006.
18. <http://lucene.apache.org/java/docs/>, Retrieved June/2006.

VIRMA: Visual Image Retrieval by Shape MAtching

G. Castellano, C. Castiello, and A.M. Fanelli

Computer Science Department, University of Bari, Via Orabona 4, 70126 Bari, Italy
{castellano, castiello, fanelli}@di.uniba.it

Abstract. The huge amount of image collections connected to multimedia applications has brought forth several approaches to content-based image retrieval, that means retrieving images based on their visual content instead of textual descriptions. In this paper, we present a system, called VIRMA (Visual Image Retrieval by Shape MAtching), which combines different techniques from Computer Vision to perform content-based image retrieval based on shape matching. The architecture of the VIRMA system is portrayed and algorithms underpinning the developed prototype are briefly described. Application of VIRMA to a database of real-world pictorial images shows its effectiveness in visual image retrieval.

1 Introduction

Recent years have seen an explosive growth in the size of digital image collections, that are being created and employed in a wide variety of applicative fields, from art to tourism, from medicine to education and entertainment. Hence, efficient and effective image retrieval from large image repositories has become an imminent research issue.

In this context, the use of traditional information retrieval methods relying on textual description of images (using keywords or free text) provides high expressive power and allows straight application of existing text retrieval software to perform image retrieval. However, there are several difficulties with this approach [10,18]. First of all, the textual description of an image is prone to be subjective, since differences in human perception makes it possible for two individuals to have diverging interpretations of the same image, especially for images with rich content. In addition, the indexer and the searcher may use synonyms or even different terms to describe the same image. This makes image retrieval based on annotation even more difficult and leads to a distinction between indexing and retrieval: while it is relatively easy to describe and index images, the search results depend on the searcher's knowledge of the indexing terms used. Moreover, some structural image characteristics (such as colours, having a broad range of different shades and intensities) are difficult to describe with words, unless using fuzzy terms like "more" or "less". Finally, textual description of an image is mainly a manual process. The potentially large amount of images in a

digital collection makes textual annotation of images a tedious and very time consuming process.

The difficulties and limitations connected with text-based techniques, especially when they are applied to large-scaled image collections, can be overcome by employing content-base image retrieval methods. This kind of techniques, rather than relying on (manual) annotation and textual descriptions, establishes indexing and retrieval of images on visual image content [8,14]. The main advantage of the content-based approach is that the extraction of visual features is predominantly an automatic process. Moreover, content-based image retrieval exploits primitive visual features that humans use to analyze and understand the content of an image, thus reducing the cognitive effort of the user in retrieving from the database results that better match user's expectation.

Several approaches have appeared in the literature that perform visual querying by examples taking into account different facets of pictorial data to express the image contents, such as colour [4,7], object shape [2,8], texture [15], or a combination of them [11,17,19]. Among these, search by matching shapes of image portions is one of the most natural way to pose a query in image databases.

In this paper we present VIRMA, a Visual Image Retrieval system that performs indexing and querying based on shape Matching. The VIRMA system provides a sample-based engine mechanism that allows the user to search the database by submitting a query that is simply represented by the sketch of a sample image. The results of the query emerge from a comparison between the submitted sample shape and the shapes of all objects appearing in the images stored into the database.

2 The VIRMA System

VIRMA is a prototype image retrieval system that combines different techniques from Computer Vision to perform Visual Image Retrieval by shape Matching. The architecture of VIRMA (as depicted in figure 1) is based on three main modules:

1. *Database Filling*. This module is devoted to collect images into the Image database and to pre-process each image so as to extract the corresponding shape content information, as described in section 2.1. Shapes of objects identified in the image are stored in a separate database, called Shape database.
2. *Database Visualization*. This module visualizes images stored in the Image database and shapes of objects stored in the Shape database;
3. *Search Engine*. This module allows the user to submit a sample image sketch as a query to interrogate the database and retrieves images containing objects whose shape matches the sketched shape according to a similarity measure, as described in section 2.2.

In the following, we describe in more detail the image storing process, performed by the Database Filling, and the image retrieval process, performed by the Search Engine.

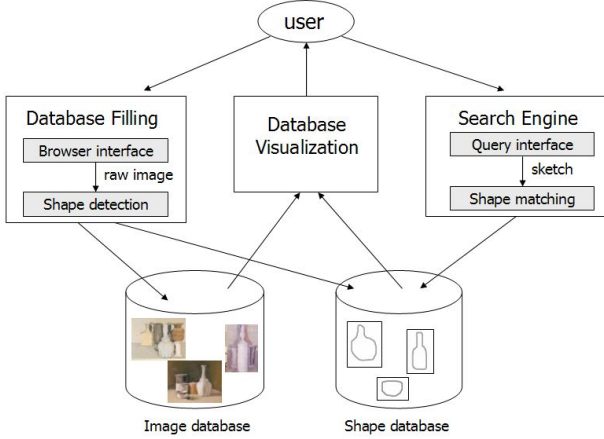


Fig. 1. Architecture of the VIRMA system

2.1 Database Filling

Each image in the database is stored as a collection of shapes representing objects contained in it. To this aim, a new image is subjected to a processing phase aimed to identify objects appearing in it. After all edges in the image have been extracted, a shape identification tool identifies different objects in the image and determines their polygonal contours. Finally, a shape simplification tool reduces the number of vertices used to represent the polygonal contour of each object, so as to decrease the amount of information needed to represent shapes to be stored in the database and to speed up the shape matching underlying retrieval. In the following, we detail each step.

Edge detection. To perform edge detection, VIRMA integrates two different tools. The first tool implements the standard edge detection method based on the Canny operator. The second edge-detection tool is based on a neuro-fuzzy approach to classify image pixels into three classes: edge, texture and regular. The neuro-fuzzy strategy, based on the correspondence between a fuzzy inference engine and a connectionist system, can learn a set of fuzzy rules that classify a single pixel g_{ij} as contour, regular or texture point, according to its edge strength values $E_{ij}^{(r)}$, $r = 1, \dots, R$, in R different scale representations. The k -th fuzzy rule is expressed in the form:

$$\text{IF } E_{ij}^{(1)} \in A_k^{(1)} \text{ AND } \dots \text{ AND } E_{ij}^{(R)} \in A_k^{(R)} \text{ THEN } g_{ij} \in C_p \text{ with degree } b_{kp} \quad (1)$$

where $A_k^{(r)}$ are fuzzy sets defined over the variables $E_{ij}^{(r)}$ using Gaussian membership functions, and b_{kp} , $p = 1, 2, 3$, are fuzzy singletons representing the degree to which a pixel belongs to one of the output classes: contour (C_1), regular (C_2) and texture (C_3). This fuzzy classification system is modelled through a three-layer

feed-forward neural network (the neuro-fuzzy network) reflecting the structure of the fuzzy rule base in its topology and the free parameters of the fuzzy rules in its adjustable weights. For mathematical details concerning the adopted learning algorithms, the reader is addressed to some other previous works of ours [6]. The advantage of using the neuro-fuzzy edge detector with respect to the standard Canny method can be appreciated experimentally. Indeed, we have observed that contours obtained by the neuro-fuzzy edge detector are much clearer (i.e. with few spurious points) and more complete (i.e. with few breaks) than those obtained by the Canny method [5]. With both methods, however, the resulting edges may contain spurious pixels that may hamper the subsequent phase of polyline construction. Thus we included in VIRMA an Edge Improvement tool that removes spurious vertices, isolated points and short edges (i.e. edges made of less than 10 points), so as to obtain a thin contour with no spurious points. Nevertheless, for complex images containing many overlapping objects, human intervention may be necessary to manually remove spurious edge points and/or draw missing parts of an object contour. This manual retouch of contours can be done through an Edge Reconstruction tool integrated in the VIRMA system.

Shape identification. After edge detection, edge pixels are analyzed by the Shape Identification tool to detect object shapes and represent them as polygonal curves. This tool is based on a particular contour following algorithm [12] that, starting from an edge pixel, follows the contour by linking adjacent edge points into a single chain, so that an object shape is reconstructed. The final result is a collection of closed curves, each curve represented by an ordered list of pixel coordinates. Hence, at the end of this process each pixel is labeled with a number representing the curve it belongs to.

Shape simplification. The last phase necessary to store image shapes in the database is the simplification of curves representing the shape of objects in the image. This is necessary not only to save space in the Shape database but also to make faster the shape matching process performed in the retrieval phase. Simplification of an object shape is achieved by reducing the number of vertices in the polyline. This is done by a simple curve evolution procedure [13] that reduces vertices iteratively. After shape simplification, each object shape identified in the image is stored in the Shape database as a list of vertices. Figure 2 shows the results of each processing step involved in the shape detection process.

2.2 Search Engine

The core of VIRMA is the Search Engine, a package of functions that allow the user to submit a sample image sketch as a query to interrogate the database. Firstly the presented sketch, which is assumed to be just the contour of a shape, is processed by the Shape Simplification tool, in order to be represented as a polygonal contour made of a reduced list of vertices. Successively, a shape matching process is applied in order to evaluate similarity between the sample shape and the shapes of all objects stored in the Shape database.

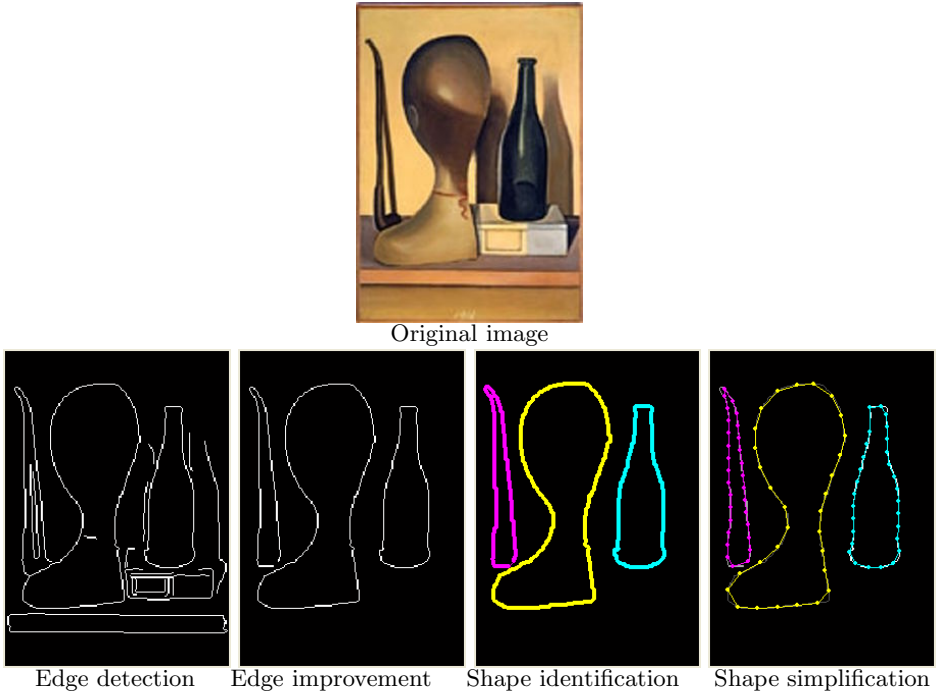


Fig. 2. The processing steps performed by VIRMA to detect object shapes in an image

To evaluate similarity we define a shape dissimilarity measure according to [1], where comparison of polygonal curves is based on the distance between their turn angle representations, also called tangent space representations or turning functions. Given a curve A expressed as a list of vertices, its turning function $\Theta_A(s)$ measures the angle of the counterclockwise tangent as a function of the arc length s , measured from some reference point on A 's boundary. By definition, the function $\Theta_A(s)$ is invariant under translation and scaling of the polygon A . Rotation of A corresponds to a shift of $\Theta_A(s)$ in the θ direction. Now, given two polygonal curves A and B (that are assumed to have the same length for simplicity), their dissimilarity can be measured by evaluating the distance between their turning functions $\Theta_A(s)$ and $\Theta_B(s)$:

$$\delta(a, B) = \|\Theta_A - \Theta_B\| = \left(\int_0^1 |\Theta_A(s) - \Theta_B(s)|^2 ds \right)^{\frac{1}{2}},$$

where $\|\cdot\|$ is the L_2 norm. Unfortunately, this distance function is sensitive to both the rotation of polygons and choice of reference point. Conversely, to achieve efficient retrieval, the measure used to evaluate similarity (or dissimilarity) between two shapes should be invariant under rotation, translation and scaling of the shapes. Hence, it makes sense to consider as distance measure, the

minimum over all possible rotation angles θ and all possible shifts t with respect to the reference point O :

$$d(A, B) = \left(\min_{\theta \in \mathbb{R}, t \in [0, 1]} \int_0^1 |\Theta_A(s+t) - \Theta_B(s) + \theta|^2 ds \right)^{\frac{1}{2}}. \quad (2)$$

To alleviate the computational burden required by computation of (2), we compute a discretization of it by considering only some values of t and θ . Of course, the quality of approximation depends on the number of such values. The rotation angle θ may range in $[0, 2\pi]$; different tests have led to consider 25 equally spaced values $\theta_i = i \cdot \frac{\pi}{12}$ for $i = 0, \dots, 24$. As concerns possible values of t (choice of reference point), we consider each vertex of the shape as a possible reference point. As stated above, each shape, after a simplification process, is represented as a list of 30 vertices, hence we consider 30 different reference points.

3 Experimental Results

For the sake of evaluation, the VIRMA system has been applied to a database composed of 20 digitalized images reproducing paintings by the Italian artist *Giorgio Morandi* [16]. All the images represent still-object pictures and therefore appear to be quite suitable for evaluating the shape detection capability and the retrieval accuracy. The effectiveness of the retrieval process has been valued both at quantitative and qualitative levels. The implementation of the VIRMA system and the sessions of experiments have been performed in the Matlab environment.

The Database Filling module of VIRMA has been run on the entire set of 20 digitalized images, in order to derive the Image and the Shape databases. The human intervention at the end of the edge-detection phase was useful to discharge from the Shape database all the wrongful forms and contours that are inevitably perceived by the system. In this way, a number of shapes were prevented to fill the database, and we were able to make room only for the consistent shapes. In a previous work of ours we have highlighted how the implemented neuro-fuzzy pixel classification is able to produce a better quality of the results, with respect to the adoption of the Canny operator [5]. For this reason, here we evaluate the VIRMA system on the basis of the Shape database derived by means of the neuro-fuzzy approach. The capability of the system in detecting shapes can be appreciated in figure 3, where the results of the detection process are reported for a sample image. It can be noticed that the neuro-fuzzy pixel classification allows a good quality of the results, which in turn implies also a reduced human intervention during the edge reconstruction phase.

With reference to the specialised literature, two levels of evaluation can be considered for rating the retrieval accuracy of the system, involving both quantitative and qualitative measures. To perform a quantitative measure of retrieval accuracy, we take into account the recall R and precision P indicators [9] that are defined in terms of relevant items and retrieved items in the context of a research process. In particular, let T represent the total number of relevant items

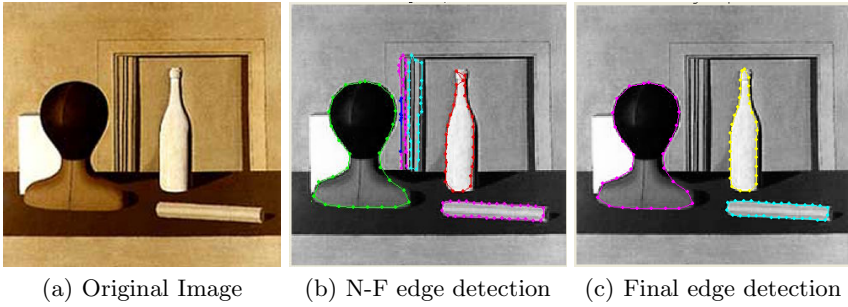


Fig. 3. The results of the shape detection process

Table 1. Values of recall and precision for 5 distinct queries

	Oblong	Squared	Squared-bottle	Round-bottle	Irregular
Recall	1	0.86	0.67	0.70	0.83
Precision	0.46	1	0.77	0.88	1

available in a query and let T_r represent the total number of retrieved items for the same query. If we indicate by R_r the total number of relevant retrieved items, then the recall and precision indicators for the results of a query can be defined as:

$$R = \frac{R_r}{T}, \quad P = \frac{R_r}{T_r}.$$

In order to evaluate the measures of recall and precision for the VIRMA system, we settled a number of 5 distinct queries, based on the discrimination of 5 prototype shapes inside the obtained Shape database. Particularly, the 5 queries refer to: 1. oblong shapes; 2. squared shapes; 3. round-bottle shapes; 4. squared-bottle shapes; 5. irregular shapes. The queries have been performed by adopting 5 prototype sketches, illustrated in figure 4. Table 1 reports the recall and precision values for each of the queries, which appear to be quite satisfactory and in some cases match the perfect rate (i.e values equal to 1). Only precision for oblong shapes has a value less than 0.5: this can be explained if we consider that elongated objects are very distributed in the dataset images, and therefore the system retrieved a great number of shapes for this particular query.

As concerning the qualitative evaluation of the retrieval results, we aimed at verifying how much the VIRMA responses to a query correspond to the human perception capabilities. For this kind of analysis we highlighted a number of 25 sample images of bottles in the Image database and 3 templates representing different bottle shapes. The highlighted images and the templates are reported in figure 5(a) and 5(b), respectively. On the basis of these images, we organized a test, involving 15 people, where every person was asked to classify the 25 sample images with respect to their similarity with each of the three templates. In this way, the testers were involved in the same task performed by the VIRMA

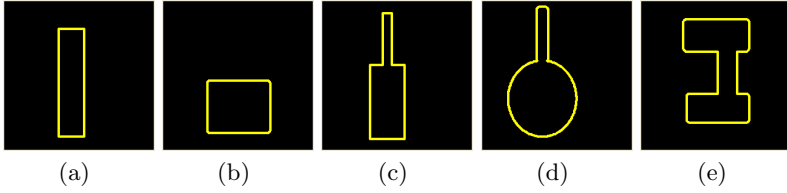


Fig. 4. The prototype sketches adopted for the quantitative evaluation of the VIRMA system: (a) Oblong shape; (b) Squared shape; (c) Squared-bottle shape; (d) Round-bottle shape; (e) Irregular shape

system. We collected from the tests the rankings $p_j(i)$ of the i th image with reference to the j th template and we evaluated the standard deviation $\sigma_j(i)$ of the ranking results. For each image i and for each rank $k = 1, \dots, 25$, we evaluated three functions $Q_j(i, k)$ ($j = 1, \dots, 3$) which measure the percentage of people who ranked the i th image in the k th position (with reference to the j th template). These functions express the similarity perception of human users, and their values should be used to perform the qualitative evaluation of the system. In practice, let $P_j(i)$ be the ranking of the i th image with reference to the j th template, as expressed by VIRMA; we consider the following function $S_j(i)$ as in [3]:

$$S_j(i) = \sum_{k=P_j(i)-\lceil \frac{\sigma_j(i)}{2} \rceil}^{k=P_j(i)+\lceil \frac{\sigma_j(i)}{2} \rceil} Q_j(i, k),$$

which computes the percentage of people who classified an image in the same rank as VIRMA or a close neighbor, defined by a window of width $\sigma_j(i)$, centered in the rank $P_j(i)$. The values of the function $S_j(i)$ for each template, averaged over the 25 considered images, are: $\bar{S}_1 = 77$, $\bar{S}_2 = 79$ and $\bar{S}_3 = 75$. These values demonstrate an agreement between the human judgment and the system capability of retrieving similar images from the Shape database.

Even if a fair comparison of the obtained results with the performances of other image retrieval systems is quite impractical (this is mostly due to the employment of different image datasets and retrieval mechanisms, together with the intrinsic subjectivity of the human judgment), the measures adopted for assessing a quantitative and qualitative evaluation allow to compare the VIRMA system with other methodologies. In particular, in [3] the same measures have been adopted and similar results have been reported, although the authors employed a database of images which is very similar, but not identical to the one we have chosen.

4 Conclusions

VIRMA represents an example of content-based image retrieval system, whose prototypic implementation has been developed in the Matlab environment.

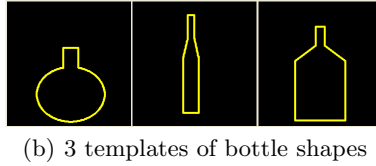
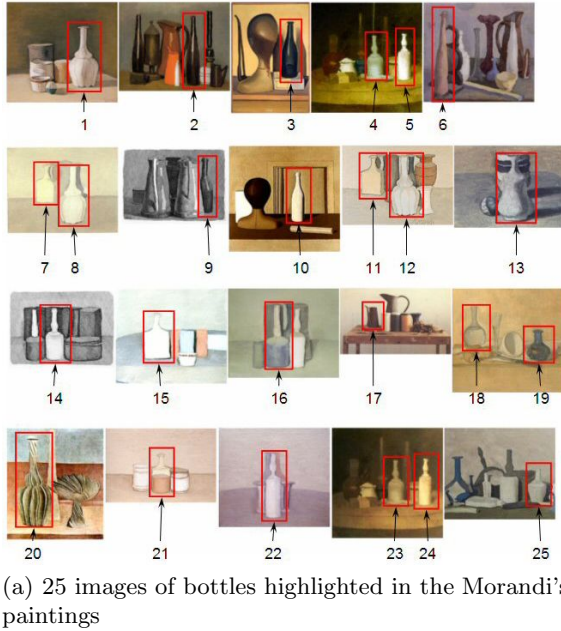


Fig. 5. The results of the shape research process for a simple object sketch

Starting from the analysis of digitalized images, the distinct modules of the system allow to collect the graphical information concerning the image dataset and to perform a retrieving process, based on shape similarity. In particular, two databases are properly derived by the system, in order to store the examined images and their shape contents, respectively. The search engine lets the user formulate a query by presenting a sample sketch image. Experimental evidence demonstrated that VIRMA is able to perform shape-based retrieving processes in agreement with the human judgment. This could pave the way for its integration in specialized multimedia systems or web applications.

As future work, we intend to investigate the possibility of combining content-based techniques and text-based methods for image retrieval. In this way, it could be possible to perform retrieving processes by operating both at low level (involving the extracted shape features composing the structure of an image) and at higher levels (involving semantic contents related to the description of an image).

References

1. Arkin, E., Chew, P., Huttenlocher, D., Kedem, K., Mitchel, J.: An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 13(3) (1991) 209–215
2. Berretti, S., D'Amico, G., Del Bimbo, A.: Shape representation by spatial partitioning for content based retrieval applications. In *proc. of IEEE International Conference on Multimedia and Expo (ICME 2004)* (2004) 791–794
3. Berretti, S., Del Bimbo, A., Pala, P.: Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Trans. on Multimedia* 2(4) (2000) 225–239
4. Binaghi, E., Gagliardi, I., Schettini, R.: Image retrieval using fuzzy evaluation of color similarity. *Int. J. Pattern Recognition and Art. Int.* 8(4) (1994) 945–968
5. Castellano, G., Castiello, C., Fanelli, A.M.: Content-based image retrieval by shape matching. In *proc. of North American Fuzzy Information Processing Society (NAFIPS 2006)*, 2006.
6. Castiello, C., Castellano, G., Caponetti, L., Fanelli, A.M.: Classifying image pixels by a neuro-fuzzy approach. In *proc. of International Fuzzy Systems Association World Congress (IFSA 2003)*, 253–256, Istanbul, Turkey, 2003
7. Del Bimbo, A., Mugnaini, M., Pala, P., Turco, F.: Visual querying by color perceptive regions. *Pattern Recognition* 31 (1998) 1241–1253
8. Del Bimbo, A., Pala, P.: Visual image retrieval by elastic matching of user sketches. *IEEE Trans. Pattern Anal. Machine Intell.* 19 (1997) 121–132
9. Eidenberger, H.: A new perspective on Visual Information Retrieval. *SPIE IS&T Electronic Imaging Conference*, San Jose, USA (2004)
10. Huang, T. S., Rui, Y.: Image Retrieval: Current Techniques, Promising Directions And Open Issues. *Journal of Visual Comm. and Image Repr.* 10(4) (1999) 39–62
11. Jain, A.K., Vailaya, A.: Image retrieval using color and shape. *Pattern Recognition* 29(8) (1996) 1233–1244
12. Kovesi, P.D.: *MATLAB and Octave Functions for Computer Vision and Image Processing*. School of Computer Science and Software Engineering, The University of Western Australia.
13. Latecki, L.J., Lakamper, R.: Shape similarity measure based on correspondence of visual parts. *IEEE Trans. on Pattern Analysis and Mach. Intell.* 22 (2000), 1–6
14. Long, F., Zhang, H.J., Feng, D.D.: *Fundamentals of Content-Based Image Retrieval*. In *Multimedia Information Retrieval and Management - Technological Fundamentals and Applications*, Feng D., Siu W.C., Zhang H.J. (Eds.), Springer, 2003
15. Liu, F., Picard, R. W.: Periodicity, directionality, and randomness-Wold features for image modeling and retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence* 18 (1996) 722–733
16. Museo Morandi, web page: <http://www.museomorandi.it/>
17. Pala, P., Santini, S.: Image retrieval by shape and texture. *Pattern Recognition* 32 (1999) 517–527
18. Santini, S., Jain, R.: Beyond Query by Example. *ACM Multimedia'98*. Bristol, UK, ACM (1998) 345–350
19. Zhou, S., Venkatesh, Y.V., Ko, C.C.: Texture retrieval using tree-structured wavelet transform. In *proc. of Int. Conference on Computer Vision, Pattern Recognition, and Image Processing (CVPRIP 2000)* 2000

Asymmetric Page Split Generalized Index Search Trees for Formal Concept Analysis

Ben Martin¹ and Peter Eklund²

¹ Information Technology and Electrical Engineering
The University of Queensland
St. Lucia QLD 4072, Australia
`monkeyiq@users.sourceforge.net`

² School of Economics and Information Systems
The University of Wollongong
Northfields Avenue, Wollongong, NSW 2522, Australia
`peklund@uow.edu.au`

Abstract. Formal Concept Analysis is an unsupervised machine learning technique that has successfully been applied to document organisation by considering documents as objects and keywords as attributes. The basic algorithms of Formal Concept Analysis then allow an intelligent information retrieval system to cluster documents according to keyword views. This paper investigates the scalability of this idea. In particular we present the results of applying spatial data structures to large datasets in formal concept analysis. Our experiments are motivated by the application of the Formal Concept Analysis idea of a virtual filesystem [11,17,15]. In particular the libferris [1] Semantic File System. This paper presents customizations to an RD-Tree Generalized Index Search Tree based index structure to better support the application of Formal Concept Analysis to large data sources.

1 Introduction: Information Retrieval and Formal Concept Analysis

Formal Concept Analysis [10] is a well understood technique of data analysis. Formal Concept Analysis takes as input a binary relation I between two sets normally referred to as the object set G and attribute set M and produces a set of “concepts” which are a minimal representation of the natural clustering of the input relation I . A concept is a pair $(A \subseteq G, B \subseteq M)$ such that A cannot be enlarged without reducing $|B|$ and vice versa. The application of Formal Concept Analysis to non binary relations, such as a table in a relational database, can be achieved by first transforming or “scaling” the input data into a binary relation [10,18].

A common approach to document and information retrieval using Formal Concept Analysis is to convert associations between many-valued attributes and objects into binary associations between the same objects G and new attributes M . For example, a many-valued attribute showing a person’s income as numeric

data may be converted into three attributes: low-income, middle-income and high-income which are then associated with the same set of people G . The binary relation I between $g \in G$ and $m \in M = \{\text{low-income, middle-income, high-income}\}$ is formed by asserting gIm depending on the level of the numeric income value of person g . The binary relation I is referred to as a formal context in Formal Concept Analysis.

This is the approach adopted in the ZIT-library application developed by Rock and Wille [20] as well as the Conceptual Email Manager [6]. The approach is mostly applied to static document collections (such as newsclassifieds) as in the program RFCA [5] but also to dynamic collections (such as email) as in MAIL-SLEUTH [2] and files in the Logical File System (LISFS) [17]. In all but the latter two the document collection and full-text keyword index are static. Thus, the FCA interface consists of a mechanism for dynamically deriving binary attributes from a static full-text index. Many-valued contexts are used to materialize formal contexts in which objects are document identifiers.

A specialised form of information retrieval system is a virtual file system [11,17,15]. The idea of using Formal Concept Analysis to generate a virtual filesystem was first proposed by using a logical generalization of Formal Concept Analysis [8,7] and in more recent work using an inverted file index and generating the lattice closure as required by merging inverted lists [17]. In such a system scalability becomes a critical concern because they deal with potentially millions of documents and hundreds/thousands of attributes. For this reason we investigate other forms of indexing data structure more fit to the systems scalability requirement and in particular an analysis of spatial indexing methods which have been applied in the libferris virtual file system [15].

2 Indexing and Scalable Knowledge Processing with Formal Concept Analysis

An index structure allows one to quickly find data by a retrieval key. For example the key might be a person's email address and the data could be a record with other information about that person. The index stores only the key or parts thereof and links to the information itself. Thus index entries are keys.

Typical Formal Concept Analysis queries seek all index entries which either (a) exactly match a given key or (b) are a super set of the given key. As an example of (b) consider Formal Concept Analysis on animal species: one concept might contain the attributes $\{\text{has-tail, has-fur}\}$, to find the objects which match this concept we will want all known objects which have *at least* these attributes but may include other attributes as well. Both of these common queries can be vastly aided with spatial indexing. Note that even exact match queries present problems for conventional B-Tree indexes due to attribute ordering in index creation [16].

A Generalized Index Search Tree [13,3] abstracts the core operations of a tree index structure into a small well defined collection of functions. A Generalized Index Search Tree is constructed from a collection of pages. A page is usually much larger than individual keys and may contain many keys. Page sizes are

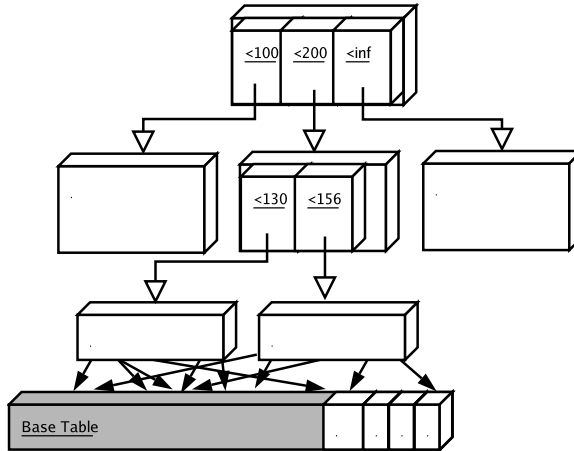


Fig. 1. Basic Generalized Index Search Tree structure. There are four internal nodes (shown at the top) and two leaf nodes (just above the base table) which contain links to the actual information that is indexed. The page size for this tree is illustrative and would normally contain hundreds/thousands of entries.

generally selected based on the hardware that the system runs on. A typical page size is 8Kb which are usually also referred to as nodes [9]. Some nodes contain links to other “child” nodes and thus form a tree. Nodes with child links are referred to as internal nodes. Children without links to other nodes – leaf nodes – contain links to the indexed data itself. An example tree is shown in Fig. 1.

The process of adding an entry to a Generalized Index Search Tree can be considered as two parts: finding the appropriate leaf node in the index, and adding an entry to that leaf. Finding the leaf node occurs in a similar manner to normal search. The difference between search and insert being when a node has multiple children which could contain the new entry a **penalty** function is used to decide which child node to insert the entry into. An entry can only be inserted into one leaf node. Typically **penalty** will select the child node which is the most closely related to the new entry to aid future searches.

When data is added to a Generalized Index Search Tree eventually a leaf node will become overfull. When a node is overfull it is typically split into two nodes: the original node and a new node. Entries are then redistributed between the original node and the new node. The creation of the new node necessitates a new entry in the original node’s parent linking the new node into the tree structure. By adding an entry to the parent node the parent itself may become overfull and thus the process of splitting nodes will continue up the Generalized Index Search Tree to the root while overfull nodes still exist.

The process of splitting a node in a Generalized Index Search Tree is delegated to the **picksplit** function. This function decides for each of the entries in the overfull node whether to store that entry into the original node or new node and provides the updated and new keys for the parent node.

It can be seen from the above description that the functions which will decide the shape of a Generalized Index Search Tree are the **penalty** and **picksplit** functions. We now consider customizations of the RD-Tree **picksplit** function for better Formal Concept Analysis performance.

3 Asymmetric Page Split

We now focus on improving the **picksplit** function and its application in Formal Concept Analysis applications.

The R-Tree [12] is a data structure that was created to allow spatial objects to be indexed effectively. Keys in an R-Tree are n -dimensional bounding boxes. The R-Tree **picksplit** first selects the two elements whose bounding box are furthest apart as keys to be placed into the parent. The remaining elements are then distributed as children of one of these new parent keys. A child is distributed into the node to which it causes the least expansion of that node's bounding box.

The RD-Tree [14] operates similarly by treating input as an n -dimensional binary spatial area. The R-Tree notion of containment is replaced by set inclusion and the bounding n -dimensional box replaced by a bounding set. The union of a collection of sets forms the bounding set. The bounding set of a child is thus defined as the union of all the elements in the child. The bounding set defined in this way preserves the “containment” notion of the R-Tree during search as a subset relation. When seeking an element which might be in a child it is sufficient to test if the sought element is a subset of the bounding set for the child to know if that subtree should be considered. The standard RD-Tree **picksplit** is based on the spatial R-Tree index structure's **picksplit**.

There is a major distinction between the standard R-Tree index keys and the RD-Tree index keys: the R-Tree index key is based around n -dimensional bounding boxes whereas the RD-Tree is based around a “bounding set”. For a given page the bounding set is simply the union of all the keys in the page. The bounding set faithfully serves the same purpose as bounding box from which it was derived; both can be treated in a similar manner as a “container” in which all keys of a child page must reside.

Any n -dimensional bounding box can always be represented as $2 \times n$ coordinates: those coordinates in n -space on two opposite sides of the bounding box. A bounding set has no such fixed representation and can require an arbitrary number of elements to represent it (up to the set cardinality). This distinction is very important, an RD-Tree based index structure needs to focus on keeping its keys small, particularly those closer to the root of the tree. Both a voluminous bounding box and a bounding set with many elements are less effective in limiting the amount of a Generalized Index Search Tree that must be searched. However, a large bounding set will also consume more of a page, limiting the branching factor of the tree.

For an index on the same information, a tree with a lower branching factor will be a deeper tree [9]. The limited branching factor index will also have more internal nodes. The efficient caching of internal nodes in a computer's RAM is

critical to index performance [9]. Having more internal nodes will decrease the effectiveness of such RAM caches.

Another critical factor in the selection of small bounding sets is that once a bounding set is selected for a parent that bounding set can never be made smaller. Further compounding this issue is that poorly chosen bounding sets at the leaf node level will eventually promote overly large bounding sets towards the root of the Generalized Index Search Tree.

Minimization of the number of elements in any bounding set should be a priority given that variation in size of the bounding set effects the overall tree branching factor. At times a *picksplit* function should favour making one of the new bounding sets slightly larger if it means a that the other bounding set can become substantially smaller.

If the page split is too asymmetric then one of the resulting pages may contain only a single element. Such an index structure may lead to many leaf pages being drastically under full and degrade overall performance. To counter this situation a minimum page fill ratio can be selected with a final element reshuffle to attempt to get closer to this fill ratio. Though it would be simple to shuffle elements until a predefined minimum page fill was achieved, it is better to try to get closer to this ratio while still minimizing the number of new elements that have to be added to the under full page's bounding set.

Thus the customized asymmetric picksplit algorithm preallocates elements to pages where they will not expand the page key, tries to minimize the expansion to one of the bounding sets, incrementally takes into account any expansion of bounding sets while distributing elements and attempts to leniently maintain a minimum page fill. The basic algorithm is shown in Fig. 2.

4 Performance Analysis

The benchmark system is an AMD XP running at 2.4GHz with 1Gb of RAM. The database is stored on a single 160Gb 7200RPM PATA disk.

Testing was performed on the Covtype database from the UCI dataset [4] and a synthetic formal context generated with the IBM synthetic data generator [19].

The implementations are as follows: **RD-Compress** is an RD-Tree using the standard *picksplit* with compression of bounding sets. **RD** is an RD-Tree using the standard *picksplit* and no compression of bounding sets. **Asym-NC** which uses an asymmetric page split algorithm. **Shuffle** which builds on **Asym-NC** by performing a shuffle after the initial allocation in an attempt to subvert the creation of drastically under full pages. Finally **Shuffle-Compress** builds on **Shuffle** by including compression of bounding sets. The compression technique used in both **Shuffle-Compress** and **RD-Compress** is identical.

Two major metrics of interest are the tree depth and the number of internal nodes in the Generalized Index Search Tree. It is unlikely that an entire Generalized Index Search Tree will be resident in RAM. Normally some of the tree can be cached in RAM for a successive search. By minimizing the number of internal nodes and the overall tree depth we can increase the chances that successive searches can find an internal node in the RAM cache.

1. Using the standard RD-Tree method select the initial left L and right R sets as new parent keys.
2. For all sets yet to be distributed, preallocate any set which is a non strict subset of either parent key $\{L, R\}$.
3. For all unallocated keys
 - (a) Test if the current key is a subset of either updated parent key, if so then allocate that key to the child page of the respective parent key.
 - (b) Attempt to minimize expansion of either parent key, when expansion has to occur prefer to expand the right parent's bounding set.
 - (c) If both parents would have to expand the same amount to cater for a key then distribute as per the normal RD-Tree method.
4. If either page is drastically under full shuffle keys into it from the other page. A page is under full if it is less than 10% utilized. Do not expand the under full page's bounding set by more than x elements. For our testing $x = 1$.

Fig. 2. Pseudo code for asymmetric page split. Preallocation will require a traversal over all keys to be distributed and set union with each key and L and R . The next step is the central part of the algorithm and only loops over keys once. The central distribution will require set unions with each key and both L and R . These can not be cached from the values computed during preallocation because L and R are incrementally expanded during this phase. The final shuffle phase potentially touches most of the keys to be distributed.

4.1 Performance on UCI Covtype Dataset

The UCI covtype database consists of 581,012 tuples with 54 columns of data. For this paper two ordinal columns were used: the aspect and elevation. A scaled table scaledcov was created from the original data source with a bit varying field containing a single bit for each formal attribute. Formal attributes were created for the most frequent 256 values for both aspect and elevation resulting in a 512 bit column. An example formal attribute would be created from a predicate like "elevation < 3144". A smaller table called mediumscaledcov was created with only the first 10,000 tuples.

The static structure of the produced index for various Generalized Index Search Tree implementations is shown in Fig. 3. As seen in Fig. 3 using an asymmetric page split can reduce the number of internal nodes in the tree by over 15% (Comparing **Asym-NC** with **RD-NC**). Although the mean leaf fill goes down for the **Shuffle-Compress** tree compared with the **RD-Compress**, there are also fewer leaf nodes in all. Considering the statistic of: leaf node count * mean leaf fill, the **Shuffle-Compress** tree has an overall reduction of over 15% compared with **RD-Compress**.

Note that compressing the bounding sets for internal nodes has a significant effect on reducing the tree depth (compare **RD-Compress** and **RD-NC**).

Queries for the extent size of each single attribute value were executed against each index. During query execution the number of internal keys read was recorded. The results for the three uncompressed index structures are shown in

Index implementation	tree depth	index size	leaf node count	internal node count	mean leaf fill
RD-Compress	7	37.7Mb	3846	977	4946
RD-NC	17	66.1Mb	3883	4582	5247
Asym-NC	14	52.3Mb	3556	3138	4800
Shuffle	13	47.6Mb	3449	2645	4597
Shuffle-Compress	6	33.6Mb	3480	825	4625

Fig. 3. Overall statistics for various Generalized Index Search Tree implementations on the scaled UCI covtype database mediumscaledcov

Table	Index implementation	tree depth	index size	leaf node count	internal node count
t100l32n10000plen7i1	RD-Compress	3	13.3Mb	1676	26
	Shuffle-Compress	3	14.2Mb	1816	21
t1000l32n10000plen7i1	RD-Compress	4	134.6Mb	16971	260
	Shuffle-Compress	4	144.3Mb	18294	182

Fig. 4. Overall statistics for various Generalized Index Search Tree implementations on the IBM data mining synthetic database

Fig. 5.(aa). It can be seen that using asymmetric page splits and post distribution shuffling lowers the number of internal keys touched.

Testing for two attribute queries was conducted by selecting two primary attributes and querying pairings of those primary attributes with each 16th attribute. For this test only the RD-Compress and Shuffle-Compress Generalized Index Search Tree are considered. Results are shown in Fig. 5.(bb).

4.2 Performance on Synthetic Data

The following use synthetic data generated with the IBM synthetic data generator [19]. Parameters include the number of transactions (ntrans), the transaction length (tlen), length of each pattern (patlen), number of patterns (npat) and number of items (nitems). The parameters were as follows ntrans=100,000 and ntrans=1,000,000, nitems=1000, tlen=32, patlen=7, npats=10000.

The static index structure is presented in Fig. 4. The Shuffle-Compress index is larger on disk and has more leaf nodes and fewer internal nodes.

As can be seen from the 100,000 transaction table shown in Fig. 5.(cc) the Shuffle-Compress index has significantly fewer internal keys touched in most queries. There are only two queries where Shuffle-Compress touches more keys than RD-Compress. For a million transaction table Shuffle-Compress always touches fewer internal keys. For the million transaction table the mean of the

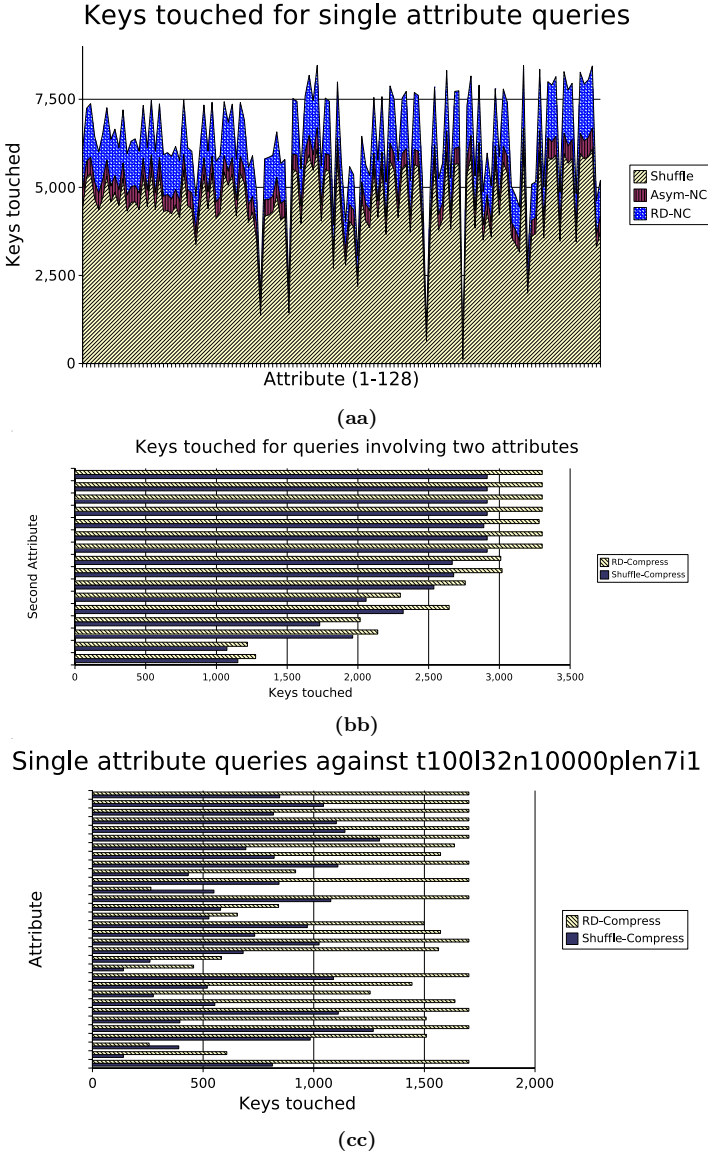


Fig. 5. Shown in (aa): Number of keys touched during single attribute extent size queries for the first 128 attributes on an uncompressed index structures. The lowest number to touched keys is always for **Shuffle**. From there upwards are: **Asym-NC** and **RD-NC**, in that order. **Shown in (bb) and cc)** The solid dark blue bar represents the **Shuffle-Compress** result while **RD-Compress** is shown in hatched yellow. **Shown in (bb):** Number of internal keys touched while querying two attributes against the **RD-Compress** and **Shuffle-Compress** Generalized Index Search Tree for every 16th other attribute with a primary attribute 25. **Shown in (cc):** Single attribute queries for the first 32 attributes in **t100l32n10000plen7i1**.

number of internal keys touched for all 32 attributes for **Shuffle-Compress** is 36% less than the same mean for **RD-Compress**.

The **Shuffle-Compress** has more leaf nodes than **RD-Compress** for both Generalized Index Search Tree. However the **Shuffle-Compress** only touched 38% the number of leaf keys that **RD-Compress** did overall. This can be explained because the **Shuffle-Compress** better clusters like keys into leaf nodes than **RD-Compress**. Also **Shuffle-Compress** being a 7% larger index size overall it has less congested leaf nodes.

5 Conclusion

This paper explores the application of a specific spatial data structure, a Generalized Index Search Tree, to the problem of query resolution using Formal Concept Analysis in the very large indexes necessary for an implementation of a semantic file system. It has been demonstrated that using an asymmetric distribution during page splitting in a Generalized Index Search Tree based on the RD-Tree can improve the resulting index structure and thus query resolution time.

The custom Generalized Index Search Tree presented offers at times a 36% drop in internal node count and in many cases a reduction of the depth of the tree over previous work [16]. These two metrics directly impact the query performance of such a tree index [9]. It should be noted that in many typical applications of Formal Concept Analysis many hundred queries are rendered against the index [16] further compounding the above performance advantages.

This ability to resolve queries in a more timely manner enables bolder applications of Formal Concept Analysis. For example, application to data sets the size of a filesystem becomes possible. Due to the indexing structure's ability to more efficiently handle high dimensional data Formal Concept Analysis can also be applied with a wider scope. For example, the ability to consider more attributes simultaneously than was previously tractable.

References

1. libferris, <http://witme.sourceforge.net/libferris.web/>. Visited Nov 2005.
2. Mail-sleuth homepage, <http://www.mail-sleuth.com/>. Visited Jan 2005.
3. Paul M. Aoki. Implementation of extended indexes in POSTGRES. *SIGIR Forum*, 25(1):2–9, 1991.
4. Blake, C., Merz, C. UCI Repository of Machine Learning Databases. [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.
5. Richard Cole and Peter Eklund. Browsing semi-structured web texts using formal concept analysis. In *Proceedings 9th International Conference on Conceptual Structures*, number 2120 in LNAI, pages 319–332. Springer Verlag, 2001.
6. Richard Cole and Gerd Stumme. Cem: A conceptual email manager. In *7th International Conference on Conceptual Structures, ICCS'2000*. Springer Verlag, 2000.

7. Sebastien Ferré and Olivier Ridoux. A file system based on concept analysis. In *Computational Logic*, pages 1033–1047, 2000.
8. Sebastien Ferré and Olivier Ridoux. A logical generalization of formal concept analysis. In Guy Mineau and Bernhard Ganter, editors, *International Conference on Conceptual Structures*, August 2000.
9. Michael J. Folk and Bill Zoelick. *File Structures*. Addison-Wesley, Reading, Massachusetts 01867, 1992.
10. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis — Mathematical Foundations*. Springer-Verlag, Berlin Heidelberg, 1999.
11. David K. Gifford, Pierre Jouvelot, Mark A. Sheldon, and James W. Jr O’Toole. Semantic file systems. In *Proceedings of 13th ACM Symposium on Operating Systems Principles*, ACM SIGOPS, pages 16–25, 1991.
12. Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In *Proc. ACM-SIGMOD International Conference on Management of Data*, Boston Mass, 1984.
13. Joseph M. Hellerstein, Jeffrey F. Naughton, and Avi Pfeffer. Generalized search trees for database systems. In Umeshwar Dayal, Peter M. D. Gray, and Shojiro Nishio, editors, *Proc. 21st Int. Conf. Very Large Data Bases, VLDB*, pages 562–573. Morgan Kaufmann, 11–15 1995.
14. Joseph M. Hellerstein and Avi Pfeifer. The RD-Tree: An Index Structure for Sets, Technical Report 1252. University of Wisconsin at Madison, October 1994.
15. Ben Martin. Formal concept analysis and semantic file systems. In Peter W. Eklund, editor, *Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, Proceedings*, volume 2961 of *Lecture Notes in Computer Science*, pages 88–95. Springer, 2004.
16. Ben Martin and Peter Eklund. In FIXME, editor, *Concept Lattices, Fourth International Conference on Formal Concept Analysis, ICFCA 2006, Proceedings*, Lecture Notes in Computer Science, page FIXME, Dresden, Germany, 2006. Springer.
17. Yoann Padioleau and Olivier Ridoux. A logic file system. In *USENIX 2003 Annual Technical Conference*, pages 99–112, 2003.
18. Susanne Prediger. Logical scaling in formal concept analysis. In *International Conference on Conceptual Structures*, pages 332–341. Springer, 1997.
19. R. Agrawal, H. Mannila, R Srikant, H. Toivonen and A. Inkeri Verkamo. Fast discovery of association rules. In U. Fayyad et al., editor, *Advances in Knowledge Discovery and Data Mining*, pages 307–328, Menlo Park CA, 1996. AAAI Press.
20. T. Rock and R. Wille. Ein TOSCANA-erkundungssystem zur literatursuche. In G. Stumme and R. Wille, editors, *Begriffliche Wissensverarbeitung: Methoden und Anwendungen*, pages 239–253, Berlin-Heidelberg, 2000. Springer-Verlag.

Blind Signal Separation of Similar Pitches and Instruments in a Noisy Polyphonic Domain

Rory A. Lewis, Xin Zhang, and Zbigniew W. Raś

University of North Carolina, KDD Laboratory, Charlotte, NC 28223, USA

Abstract. In our continuing work on "Blind Signal Separation" this paper focuses on extending our previous work [1] by creating a data set that can successfully perform blind separation of polyphonic signals containing similar instruments playing similar notes in a noisy environment. Upon isolating and subtracting the dominant signal from a base signal containing varying types and amounts of noise, even though we purposefully excluded any identical matches in the dataset, the signal separation system successfully built a resulting foreign set of synthesized sounds that the classifier correctly recognized. Herein, this paper presents a system that classifies and separates two harmonic signals with added noise. This novel methodology incorporates Knowledge Discovery, MPEG7-based segmentation and Inverse Fourier Transforms.

1 Introduction

Blind Signal Separation (BSS) and Blind Audio Source Separation (BASS) are the subjects of intense work in the field of Music Information Retrieval. This paper is an extension of previous music information retrieval work wherein we separated harmonic signals of musical instruments from a polyphonic domain. [1] In this paper we move forward and simulate some of the complexities of real world polyphonic blind source separation. Particularly, involving our previous work to recognize and synthesis signals in order to extract proper parameters from a source containing two very similar sounds in a polluted environment containing varying degrees of noise.

1.1 The BSS Cocktail Party

In 1986, Jutten and Herault proposed the concept of Blind Signal Separation by capturing clean individual signals from unknown, noisy signals containing multiple overlapping signals [2]. Jutten's recursive neural network found clean signals based on the assumption that the noisy source signals were statistically independent. Researchers in the field began to refer to this noise as the *cocktail party* property, as in the undefinable buzz of incoherent sounds present at a large cocktail party. Simultaneously, Independent Component Analysis (ICA) evolved as a statistical tool that expressed a set of multidimensional observations as a combination of unknown latent variables sometimes called dormant signals. ICA reconstructs dormant signals by representing them as a set of hypothesized

independent sequences where k = the number of unknown independent mixtures from the unobserved independent source signals:

$$x = f(\Theta, s), \quad (1)$$

where $x = (x_1, x_2, \dots, x_m)$ is an observed vector and f is a general unknown function with parameters Θ [3] that operates the variables listed in the vector $s = (s_1, \dots, s_n)$

$$s(t) = [s_1(t), \dots, s_k(t)]^T. \quad (2)$$

Here a data vector $x(t)$ is observed at each time point t , such that given any multivariate data, ICA can decorrelate the original noisy signal and produce a clean linear co-ordinate system using:

$$x(t) = \mathbf{A}s(t), \quad (3)$$

where $\mathbf{A}$ is a $n \times k$ full rank scalar matrix. Algorithms that analyze polyphonic time-invariant music signals systems operate in either the time domain [4], the frequency domain [5] or both the time and frequency domains simultaneously [6]. Kostek takes a different approach and instead divides BSS algorithms into either those operating on multichannel or single channel sources. Multichannel sources detect signals of various sensors whereas single channel sources are typically harmonic [7]. For clarity, let it be said that experiments provided herein switch between the time and frequency domain, but more importantly, per Kostek's approach, our experiments fall into the multichannel category because, at this point of experimentation two harmonic signals are presented for BSS. In the future, a polyphonic signal containing a harmonic and a percussive may be presented.

1.2 Art Leading Up to BSS in MIR, a Brief Review

In 2000, Fujinaga and MacMillan created a system recognizing orchestral instruments using an exemplar-based learning system that incorporated a k nearest neighbor classifier (k-NNC). [8] Also, in 2000, Eronen and Klapuri created a musical instrument recognition system that modeled the temporal and spectral characteristics of sound signals [9] that measured the features of acoustic signals. The Eronen system was a step forward in BSS because the system was pitch independent and it successfully isolated tones of musical instruments using the full pitch range of 30 orchestral instruments played with different articulations. In 2001 Zhang constructed a multi-stage system that segmented music into individual notes and estimated the harmonic partial estimation from a polyphonic source [10]. In 2002, Wiczkowska, collaborated with Slezak, Wróblewski and Synak [11] and used MPEG-7 based features to create a testing database for training classifiers used to identify musical instrument sounds. Her results showed that the kNNC classifier outperformed, by far, the rough set classifiers. In 2003, Eronen [12] and Agostini [13] both tested, in separate tests, the viability of using decision tree classifiers in music information retrieval.

In 2004, Kostek developed a 3-stage classification system that successfully identified up to twelve instruments played under a diverse range of articulations

[14]. The manner in which Kostek designed her stages of signal preprocessing, feature extraction and classification may prove to be the standard in BSS MIR. In the preprocessing stage Kostek incorporated 1) the average magnitude difference function and 2) Schroeder's histogram for purposes of pitch detection. Her feature extraction stage extracts three distinct sets of features: Fourteen FFT based features, MPEG-7 standard feature parameters, and wavelet analysis.

2 Experiments

To further develop previous research where we dissimilar sounds, here we used two very similarly timbered and pitched instruments. In the previous experiments we used 4 separate versions of a polyphonic source containing two harmonic continuous signals obtained from the McGill University Masters Samples (MUMs) CDs. The first raw sample contained a C at octave 5 played on a nine foot Steinway, recorded at 44,100HZ, in 16-bit stereo and the second raw sample contained an A at octave 3 played on a B $\flat$ Clarinet, recorded at 44,100HZ, in 16-bit stereo. We then created four more samples created from mixes of the first two raw samples.

In this paper's experiments we again used Sony's Sound Forge 8.0 for mixing sounds. Two base instruments used to create a mix, in terms of their timbre, are much closer in this paper than in the previous one [1]. The first raw sample contained a C at octave 4 played on a violin by a bow, played with a vibrato effect. The second raw sample contained a quite similar sounding C at octave 5 played on a flute with a fluttering technique. Both raw sounds were recorded at 44,100HZ, in 16-bit stereo and obtained from the McGill University Masters Samples (MUMs) CDs.

2.1 Noise Variations

In order to further simulate real world environments we decided to pollute the data set with noise. First we constructed a noise data set based off of four primary noises. The first noise contained sounds recorded at a museum containing footsteps, clatter, muffled noises and talking. The second noise contained the sound a very strong wind makes on as it gusts, the pitches varied and so did the intensity of the gusts. The third noise contained the sounds an old clattering sputtering air conditioner made, it included the sound of the air, the vibrations of the tin and the engine as well as the old motor straining to keep turning. The last noise contained the clanking and noise that a factory steam engine made. One can hear the steam, the pistons, and the large gears all grinding to make a terrible noise. We then mixed, using Sony's Sound Forge software, the four aforementioned raw noises to produce ten additional noise combinations. These combinations included 01:02, 01:02:03, 01:02:03:04, 01:03:04, 02:03, 02:03:04, 01:02:04, 01:03, 01:04 and 03:04. To create the sound data set, the team mixed these 10 new noises and 4 original noises with 4C violin 5C flute using Sony's Sound Forge 8.0 for mixing sounds (see Tables 1 and 2).

2.2 Creating a Real-World Training and Testing Sets

Creating a real world data set for training is the essence of the paper. In the real world, polyphonic recording invariably contain pollutants in the form of noise. Taking the created noises, explained in the previous section, the team constructed a set of sounds for training/testing as follows. We randomly selected four sets into the training set. Essentially, as in the real world a training set will have, first, noise and probably never contain the exact signal in the training set. We used WEKA for all classifications. Using similar sounds, varying noise and no exact match in the training data set achieved our real world environment as shown in Tables 1 and 2.

Table 1. Training Set: Sample Sound Mixes: Training Set Preparation

Harmonics	Noise
4C violin	5C flute 01
4C violin	5C flute 01:02
4C violin	5C flute 01:02:03
4C violin	5C flute 01:02:03:04
4C violin	5C flute 01:03:04
4C violin	5C flute 02
4C violin	5C flute 02:03
4C violin	5C flute 02:03:04
4C violin	5C flute 03
4C violin	5C flute 04

Table 2. Sample Sound Mixes: Testing Set Preparation

Harmonics	Noise
4C violin	5C flute 01:02:04
4C violin	5C flute 01:03
4C violin	5C flute 01:04
4C violin	5C flute 03:04

Upon isolating and subtracting 5C flute signal from the mix of all 10 sounds in the Training Set (see [1]), we obtained a new dataset of 10 samples of 4C violin mixed with different types of noises as shown in Table 3. Now, we extended this set by adding to it 25 varying pitches of a Steinway Piano taken from the MUM database. This new dataset was used for training and the one shown in Table 4 for testing.

2.3 Classifiers

We used four classifiers in our investigations on musical instrument recognition: Tree J48, Logistic Regression Model, Bayesian Network, and Locally Weighted Learning.

Bayesian Network. A set of variable nodes with a set of dependencies called edges that exist between the variables and a set of probability distribution functions for each variable. The nodes represent the random variables while the arrows are the directed edges between pairs of nodes. This approach has been successfully applied to speech recognition [15], [16].

Table 3. Sample Training Set: Resultant Sounds by Sound Separation

Harmonics Noise	
4C violin	01
4C violin	01:02
4C♯ piano	clean
4C♯ piano	clean

Table 4. Sample Testing Set: Resultant Sounds by Sound Separation

Harmonics Noise	
4C violin	01:02:04
4C violin	01:03
4C violin	01:04
4C violin	03:04

Tree J48. Decision Tree-J48 is a supervised classification algorithm, which has been extensively used for machine learning and pattern recognition [17]. Tree-J48 is normally constructed top-down, where parent nodes represent conditional attributes and leaf nodes represent decision outcomes. It first chooses a most informative attribute that can best differentiate the data set; it then creates branches for each interval of the attribute where instances are divided into groups, until instances are clearly separated in terms of the decision attribute; finally it tests the tree by new instances in a test data set.

Logistic Regression Model. Logistic regression model is a popular statistical approach of analyzing multinomial response variables, since it does not assume normally distributed conditional attributes which can be continuous, discrete, dichotomous or a mix of any of these; it can handle nonlinear relationships between the decision attribute and the conditional attributes. It has been widely used to correctly predict the category of outcome for new instances by maximum likelihood estimation using the most economical model [19].

Locally Weighted Learning. Locally Weighted Learning is a well-known lazy learning algorithm for pattern recognition. It votes on the prediction based on a set of nearest neighbors (instances) of the new instance, where relevance is measured by a distance function, typically a Euclidean Distance Function, between the query instance and the neighbor instance. The local model consists of a structural and a parametric identification, which involve parameter optimization and selection [20].

3 Experimental Parameters

3.1 MPEG-7 Features

In considering the use of MPEG-7, the authors recognized that a sound segment containing musical instruments may have three states: transient, quasi-steady and decay. Identifying the boundary of the transient state enables accurate timber recognition. Wierzchowska proposed a timbre detection system [21] where she splits each sound segment into 7 equal intervals. Because different instruments require different lengths, we use a new approach to look at the time it

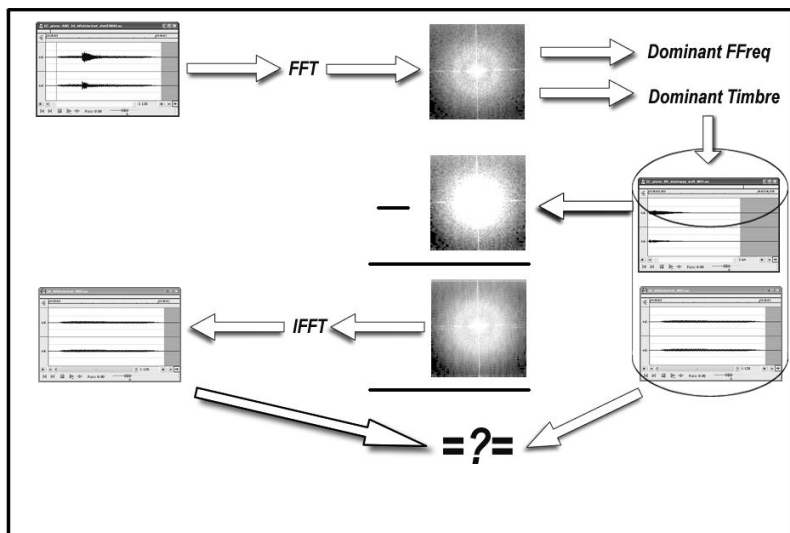


Fig. 1. Procedure for subtracting an extracted signal's FFT from the source FFT

takes for the transient duration to reach the quasi-steady state of the fundamental frequency. It is estimated by computing the local cross-correlation function of the sound object, and the mean time to reach the maximum within each frame. The classifiers we built for training/testing are based on the following MPEG-7 descriptors:

AudioSpectrumCentroid. It is a description of the center of gravity of the log-frequency power spectrum. Spectrum centroid is an economical description of the shape of the power spectrum. It indicates whether the power spectrum is dominated by low or high frequencies and, additionally, it is correlated with a major perceptual dimension of timbre; i.e. sharpness. To extract the spectrum centroid: 1. Calculate the power spectrum coefficients; 2. Power spectrum coefficients below 62.5 Hz are replaced by a single coefficient, with power equal to their sum and a nominal frequency of 31.25 Hz; 3. Frequencies of all coefficients are scaled to an octave scale anchored at 1 kHz.

AudioSpectrumSpread. It is a description of the spread of the log-frequency power spectrum. Spectrum spread is an economical descriptor of the shape of the power spectrum that indicates whether it is concentrated in the vicinity of its centroid, or else spread out over the spectrum. It allows differentiating between tone-like and noise-like sounds. To extract the spectrum Spread, we calculate the spectrum spread as the RMS deviation with respect to the centroid, on an octave scale.

HarmonicSpectralCentroid. It is computed as the average over the sound segment duration of the instantaneous HarmonicSpectralCentroid within a running window. The instantaneous HarmonicSpectralCentroid is computed as the

amplitude (linear scale) weighted mean of the harmonic peaks of the spectrum. To extract the Harmonic Spectral Centroid, 1. Estimate the harmonic peaks over the sound segment. 2. Calculate the instantaneous HarmonicSpectralCentroid. 3. Calculate the average HarmonicSpectralCentroid for the sound segment.

HarmonicSpectralDeviation. It is computed as the average over the sound segment duration of the instantaneous HarmonicSpectralDeviation within a running window which is computed as the spectral deviation of log-amplitude components from a global spectral envelope. The Harmonic Spectral Deviation is extracted using the following algorithm 1. Estimate the harmonic peaks over the sound segment. 2. Estimate the spectral envelope. 3. Calculate the instantaneous HarmonicSpectralDeviation. 4. Calculate the average HarmonicSpectralDeviation for the sound segment.

HarmonicSpectralSpread. It is computed as the average over the sound segment duration of the instantaneous HarmonicSpectralSpread within a running window computed as the amplitude weighted standard deviation of the harmonic peaks of the spectrum, normalized by the instantaneous HarmonicSpectralCentroid. It is extracted using the following algorithm 1. Estimate the harmonic peaks over the sound segment. 2. Estimate the instantaneous HarmonicSpectralCentroid. 3. Calculate the instantaneous HarmonicSpectralSpread for each frame. 4. Calculate the average HarmonicSpectralSpread for each sound segment.

HarmonicSpectralVariation. It is the mean over the sound segment duration of the instantaneous HarmonicSpectralVariation. The instantaneous HarmonicSpectralVariation is defined as the normalized correlation between the amplitude of the harmonic peaks of two adjacent frames. It is extracted using the following algorithm. 1. Estimate the harmonic peaks over the sound segment. 2. Calculate the instantaneous HarmonicSpectralVariation each frame. 3. Calculate the HarmonicSpectralVariation for the sound segment.

4 Experimental Procedures

In this research, we applied four different types of noise to built our testing dataset of 14 sounds described in Tables 1, 2. The second table represents randomly selected four of the resultant new sounds for testing. This testing consisted of binary classification of violin against a 25 varying pitches of a Steinway Piano taken from the MUM database. We used the remaining sounds in 3 for training. Essentially, we now had a training set that was void of any of the combination in the training set 4. In analyzing the results we found that the system correctly classified all the synthesized using four different classifiers. We did observe however that the original sound of violin from the MUMs database was incorrectly classified, without other original violin sounds from MUMs in the training set, by three of the classifiers: Tree J48, Logistic Regression Model, and Bayesian Network. Thus, in this research, we conclude that the original sounds from MUMs and the synthesized sound, which were produced by our subtraction algorithm, can form a robust database, which can represent similar sounds

of recordings from different sources. We consider the transient duration as the time to reach the quasi-steady state of fundamental frequency. Thus we only apply it to the harmonic descriptors, since in this duration the sound contains more timbre information than pitch information of the note, which is highly relevant to the fundamental frequency. The fundamental frequency is estimated by first computing the local cross-correlation function of the sound object, and then computing mean time to reach its maximum within each frame, and finally choosing the most frequently appearing resultant frequency in the quasi-steady status. In each frame i , the fundamental frequency is calculated in this form:

$$f(i) = \frac{S_r}{K_i/n_i} \tag{4}$$

where S_r is the sample Frequency, n is the total number of $r(i, k)$'s local valleys across zero, where $k \in [1, K_i]$. See formula. K_i is estimated by k as the maximum fundamental period by the following formula, where $r(i, k)$ reaches its maximum value. ω is the maximum fundamental period expected.

$$r(i, k) = \sum_{j=1}^{m+n-1} s(j)s(j-k) \bigg/ \sqrt{\sum_{j=m}^{m+n-1} s(j-k)^2 \sum_{j=m}^{m+n-1} s(j)^2}, \quad k \in [1, S_r \times \omega] \tag{5}$$

Table 5. Overall Accuracy Results Tree J8, Logistic, Local Weighted Learning, Bayesian Network

	<i>TreeJ48</i>	<i>Logistic</i>	<i>LWL</i>	<i>BayesianNetwork</i>
Accuracy	83.33%	83.33%	100%	83.33%

Table 6. Individual Results.

	<i>TreeJ48</i>	<i>Logistic</i>	<i>LWL</i>	<i>BayesianNetwork</i>
4C violin noise 01 02 04	Yes	Yes	Yes	Yes
4C violin noise 01 03	Yes	Yes	Yes	Yes
4C violin noise 01 04	Yes	Yes	Yes	Yes
4C violin noise 03 04	No	No	Yes	No
Subtracted Piano Sounds	Yes	Yes	Yes	Yes

5 Conclusion

Automatic sound indexing should allow labeling sound segments with instrument names. In our research, we start with the singular, homophonic sounds of musical instruments, and then extend our investigations to simultaneous sounds. This paper is focussing on the mix of two instruments sounds, the construction of

successful classifiers for identifying them, and developing a new theory which we can easily extend to the mix of several sounds. Knowledge discovery techniques are applied at this stage of research. First of all, we discover rules that recognize various musical instruments. Next, we apply rules from the obtained set, one by one, to unknown sounds. By identifying so called supporting rules, we are able to point out which instrument is playing (or is dominating) in the given segment, and in what time moments this instrument starts and ends playing.

Acknowledgements

This research is supported by the National Science Foundation under grant IIS-0414815.

References

- [1] Lewis R., Zhang X., Ras, Z.: "A Knowledge Discovery Model of Identifying Musical Pitches and Instrumentations in Polyphonic Sounds.", Special Issue, International Journal of Engineering Applications of Artificial Intelligence, 2007, will appear
- [2] Herault J.,Jutten C.: "Space or time adaptive signal processing by neural network models." American Institute of Physics **Neural Networks for Computing. Vol. 151 Editor: J. S. Denker New York 3rd**(1986)206–211
- [3] Bingham, E: "Advances In Independent Component Analysis With Applications To Data Mining" Helsinki University of Technology: PhD Dissertation, **Helsinki University of Technology.** (2003) 07–11
- [4] Amari, S. et al: "Multichannel blind deconvolution and equalization using the natural gradient" Proc. IEEE Workshop. , **Signal Processing Advances in Wireless Comm.** (1997) 101–104
- [5] Smaragdis, P. et al: "Blind separation of convolved mixtures in the frequency domain" Proc. IEEE Workshop. , **IEEE Procedures Neurocomputing: vol: 22** (1998) 21–34
- [6] Lambert, R. H and Bell A.J.: "Blind separation of multiple speakers in a multipath environment", **IEEE Proc. ICASSP: April** (1997) 423–426
- [7] Kostek, B.et al: "Estimation of Musical Sound Separation Algorithm Effectiveness Employing Neural Networks" Journal of Intelligent Information Systems, **SpringerHoughton Mifflin Company May, Vol:24 number 2/3** (2005) 133–135
- [8] Fujinaga, I and MacMillan, K: "Realtime recognition of orchestral instruments" Proceedings of the International Computer Music Conference - Best Presentation Award, **SpringerHoughton Mifflin Company** (2000) 141–143
- [9] Eronen, A., Klapuri, A.: "Musical Instrument Recognition Using Cepstral Coefficients and Temporal Features" In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, **ICASSP SpringerHoughton Mifflin Company**(2000) 753–756
- [10] Zhang, T.: "Instrument classification in polyphonic music based on timbre analysis" SPIE's Conference on Internet Multimedia Management Systems **II part of ITCom'01, Denver, Aug. 4519** (2001) 136– 147

- [11] Slezak, D., Synak, P., Wieczorkowska, A. and Wroblewski, J.: "KDD-based approach to musical instrument sound recognition." Hacid, M.S., Ras,Z.W., Zighed, D.A., Kodratoff Y. (eds.): **Foundations of Intelligent Systems.Proc. of 13th Symposium ISMIS 2002, Lyon, Franc 4519 Berlin, Heidelberg Springer-Verlag** (2002) 28– 36
- [12] Eronen, A.: "Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs" In Proceedings of Processing and its Applications, ISSPA 2003, Paris, France, 1-4 July, **Proceedings of the Seventh International Symposium on Signal Processing**(2003) 133–136
- [13] Agostini. M, Longar M and Pollastri, E.: "Musical Instrument Timbres Classification with Spectral Features" **EURASIP Journal on Applied Signal Processing, volume issue 1** (2003)5–14
- [14] Kostek, B.: "Musical Instrument Classification and Duet Analysis Employing Music Information Retrieval Techniques." **Proc. of the IEEE.92(4)** (2004)712–729
- [15] Zweig, O.: "Speech Recognition with Dynamic Bayesian Networks" Ph.D. dissertation, **Univ. of California, Berkeley, CA.** (1998)
- [16] Livescu,K., Glass, J., and Bilmes, J.: "Hidden Feature Models for Speech Recognition Using Dynamic Bayesian Network" **Geneva, Switzerland Proc. Eurospeech September** (2003) 2529-2532
- [17] Quinlan,J.: "C4.5: Programs for Machine Learning" **Morgan Kaufman, San Mateo CA** (1993)
- [18] Wieczorkowska, A.: "Classification of musical instrument sounds using decision trees" **8th International Symposium on Sound Engineering and Mastering, ISSEM'99** (1999) 225– 230
- [19] le Cessie, S., and van Houwelingen, J.: "Ridge Estimators in Logistic Regression" **Applied Statistics, 41, no. 1** (1992)191–201
- [20] Atkeson C., Moore, A., and Schaal, S.: "Locally Weighted Learning for Control" **Artificial Intelligence Review, 11 no. 1-5 Feb.** (1997) 11–73
- [21] Wieczorkowska A., et al.: "Application of Temporal Descriptors to Musical Instrument Sound" **Journal of Intelligent Information Systems, Integrating Artificial Intelligence and Database Technologies, volume 21, no. 1, July** (2003)

Score Distribution Approach to Automatic Kernel Selection for Image Retrieval Systems

Anca Doloc-Mihu and Vijay V. Raghavan

University of Louisiana at Lafayette, Lafayette, LA 70504, USA
anca@louisiana.edu, raghavan@cacs.louisiana.edu
<http://www.cacs.louisiana.edu/~axd9917>

Abstract. This paper introduces a kernel selection method to automatically choose the best kernel type for a query by using the score distributions of the relevant and non-relevant images given by user as feedback. When applied to our data, the method selects the same best kernel (out of the 12 tried kernels) for a particular query as the kernel obtained from our extensive experimental results.

1 Introduction

This paper contributes to the design and evaluation of learning methods employed in Web-based Adaptive Image Retrieval System (AIRS) in order to maximize user benefits. Here, we focus on kernel-based learning methods. Specifically, the goal of this paper is to investigate the selection of the kernel type for a Web-based AIRS.

In a Web-based Image Retrieval System, the goal is to answer as best (fast and accurate) as possible to the user's request, given here as query image(s) represented by color histograms. However, a characteristic of these colors is that they are not independent, but correlated. More, the interaction between them is stronger for some queries (images) than for other queries.

In the former case, a more complex, possibly non-linear, kernel has to be used, whereas in the latter case, a linear kernel may be sufficient. Since we are dealing with real images, and therefore, with complex queries, it is expected that we need to use non-linear kernels to achieve good retrieval results [6]. However, in the literature, "there are currently no techniques available to learn the form of the kernel" [4].

Methods like Relevance Vector Machines [12] assume distribution of data, which might fit or not a real image collection. In our previous work [7], we used the score distribution models briefly presented in Section 2.1. Instead, in here, we propose a kernel selection method based on the relevance feedback information given by user. Our method computes the score distributions based on the information given by user as feedback (relevant and non-relevant images), without using any particular model for these distributions.

Based on the Kernel Rocchio [1] learning method, several kernels having polynomial and Gaussian Radial Basis Function (RBF) like forms (6 polynomials and

6 RBFs) are applied to general images represented by color histograms in RGB and HSV color spaces. We implement and test these kernels on two image collections of sizes 5000 and 10000. From the plots of these results, we select the best (effective and efficient) kernel desired to be used for each query.

Then, we applied our method to these collections for the same queries. For each query, the method selects the same best kernel (out of the 12 tried kernels) for a particular query as the kernel obtained from our extensive experimental results.

The paper organization is as follows. Section 2 presents a new method for automatically selecting the kernel to be used by the Kernel Rocchio learning method in order to improve the performance of an adaptive image retrieval system (AIRS). Section 3 reports experimental results. Finally, Section 4 concludes the paper.

2 Score Distribution Approach for Kernel Type Selection

Researchers [9,10,8] modeled the score distributions of search engines for relevant and non-relevant documents by using a normal and an exponential distribution, respectively. Based on this idea, in this section, we propose a procedure based on score distributions to automatically select the kernel type for an AIRS.

2.1 Mathematical Model of Image Score Distributions

The score distributions of the relevant documents suggested by researchers [10] are modeled as

$$P(score|P = R) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(score-\mu)^2}{2\sigma^2}\right),$$

where μ is the mean, and σ is the variance of the Gaussian distribution. The score distributions of the non-relevant documents are modeled as

$$P(score|P = NR) = \lambda \exp(-\lambda * score),$$

where λ is the mean for the exponential distribution.

Our experimental results (Section 3.1) support this mathematical models of the score distributions, with some approximation. Figure 1 illustrates how the score distributions fit the top 300 images for one query for 6 kernels.

As an observation, the score distributions are different for the different kernel types used in the experiments (Figure 1). This motivates us to seek for a method to select the kernel type by using these differences between the score distributions.

2.2 Kernel-Based Retrieval

In Image Retrieval, the user searches a large collection of images, for instance, that are similar to a specified query. The search is based on the similarities of

the image attributes (or features) such as colors. Then, a linear retrieval form [2,1] matches image queries against the images from collection

$$F : R^N \times R^N \rightarrow R, F(P, Q) = P^t Q,$$

where query image Q contains the features desired by user, and t stands for the transpose of vector P . The bigger the value of the function F applied to a query Q and an image P , the better the match between the query image Q and the collection image P , or, in other words, the closer the two images. For representing images by color, we use the histogram representations in RGB and HSV color spaces.

Recently, image retrieval systems start using different learning methods for improving the retrieval results. In this work, we use the **Kernel Rocchio** method [1] for learning, which combines the simplicity of Rocchio's method with the power of the kernel method to achieve improved results. For any image P_k from the collection, the algorithm computes its retrieval status values (RSV) as:

$$RSV(\phi(P_k)) = \frac{1}{|R|} \sum_{P_i \in R} \frac{K(P_i, P_k)}{\sqrt{K(P_i, P_i) \cdot K(P_k, P_k)}} - \frac{1}{|\bar{R}|} \sum_{P_j \in \bar{R}} \frac{K(P_j, P_k)}{\sqrt{K(P_j, P_j) \cdot K(P_k, P_k)}}, \quad (1)$$

where R is the set of relevant images and $\bar{R}$ is the set of the non-relevant images.

The kernel method constitutes a very powerful tool for the construction of learning algorithm by providing a way for obtaining non-linear algorithms from algorithms previously restricted to handling only linearly separable datasets.

In this work, the general form of the polynomial kernels [4,1] is given by

$$K(x, y) = (\langle x, y \rangle)^d, d > 0, \quad (2)$$

where $\langle \cdot, \cdot \rangle$ is the inner product between vectors x and y . The general form of the radial kernels is given by

$$K(x, y) = \exp \left(- \frac{\left(\sum_{i=1}^N |x_i^a - y_i^a|^b \right)^c}{2\sigma^2} \right), \sigma \in R^+. \quad (3)$$

2.3 Kernel Selection Approach Based on Image Score Distributions

In image retrieval, the goal is to retrieve as many relevant images and as few non-relevant images as possible. This means, we wish to have a retrieval system capable to rank all the relevant images before the non-relevant ones. In other words, the scores of the relevant images should be as higher as possible, whereas the scores of the non-relevant images should be as lower as possible.

In our system, these scores are computed by using the different kernel types. Then, logically, the best kernel is the one that is able to distribute the scores

of the images, such that the relevant images have higher scores than the non-relevant ones. Therefore, we suggest the following procedure for selecting the kernel type:

1. after each feedback step use the relevance information given by user to compute for each kernel type K_i the score distribution of the relevant images (ScR_i), and the score distribution of the non-relevant images ($ScNR_i$), here, for $s = 10$ intervals.
2. select as the best kernel $K_i = \min_{K_j} \left(\sum_{s=1}^{10} \min \{ScR_{js}, ScNR_{js}\} \right)$.

Our method of selecting the best kernel tries to fit the score distributions of both relevant and non-relevant images, such that there are as many as possible relevant images with high scores grouped towards the right half of the plot (see Figure 1), and just a few relevant images grouped in the left half side, and vice-versa for the non-relevant images, i.e. there is a good separation between the relevant and non-relevant images. For this, we penalize the relevant images with low scores and the non-relevant images with high scores, and then, we select the best kernel as the one with the smallest penalty (error).

As Zhang and Callan noticed [13], the model is biased, especially for low scoring images, which do not occur between the top (here) 300 images. However, for high scoring images, the model offers a relatively good estimation [13,9].

However, the above procedure can be applied after each feedback step to select the best kernel type to be used for the following feedback step. Since the procedure is based on the feedback information, it follows that the method might work better when there is more feedback information. Therefore, we suggest to apply the method when using the feedback information accumulated from all previous feedback steps.

As mentioned, previous work [13,9] used different distributions to model the scores. Then, these distributions were used to predict the behavior of the retrieval model. Instead, in here, we just compute the score distributions based on the information given by user as feedback (relevant and non-relevant images), without using any particular model for these distributions. That is, we do not make any assumption about the behavior of these score distributions.

On the other hand, if we want a way of selecting the best kernel for a given query in advance, we can still use the above procedure. That is, after some feedback we select the best kernel as given by our method, and then, we use the normal and exponential distributions to predict the behavior of the model for the relevant and non-relevant images, respectively. Next, we can calculate the overall (predicted) score distributions for each kernel type by using these models. Finally, we select the kernel that shows the highest (predicted) score distribution of the relevant images and the lowest (predicted) score distribution of the non-relevant images. Note that this procedure uses the score distribution models proposed in the literature for text retrieval [10].

As a conclusion, our method for selecting the kernel type for a particular query is based on score distributions, which are obtained via feedback from user and can be calculated automatically by the system at each feedback step. Next section reports our experimental results.

3 Experiments and Results

3.1 Experimental Setup

For our experiments, we use two test collections of sizes 5000 and 10000, each including 100 relevant images, for each query image. For convenience, we name these sets as 5000_100 and 10000_100. All image collections are quantized to 256 colors in RGB [15] and 166 colors in HSV [14]. We use the same set of 10 images as queries ($Q_1, Q_2, \dots, Q_{10}$) for each experiment.

For evaluation purposes, we use the Test and Control method. The process of obtaining the training and testing sets is described in details in [6]. For each test collection, we create a training set of 300 randomly distributed images. The tests are performed on the the test set. The number of the relevant images within the training and the testing sets for each query is given in Table 1.

Table 1. Number of the relevant images within the training and testing sets

5000_100	Training	Test	10000_100	Training	Test
Q_1	9	53	Q_1	4	62
Q_2	6	50	Q_2	0	49
Q_3	4	51	Q_3	4	52
Q_4	8	56	Q_4	7	46
Q_5	7	43	Q_5	7	47
Q_6	5	56	Q_6	7	46
Q_7	4	56	Q_7	3	44
Q_8	4	48	Q_8	4	46
Q_9	5	48	Q_9	1	49
Q_{10}	6	50	Q_{10}	2	48

We perform the experiments for a set of 12 kernels: 6 polynomials and 6 radial basis, with general forms given by Equations (2) and (3), respectively. The values of the parameters (a, b, c , and d) and the names of the kernels used in the experiments are presented in Table 2. In all experiments presented in this work, $\sigma = 1$.

3.2 Discussion of the Results

Figure 1 illustrates the score distributions of the top 300 images for query Q_1 for 6 kernels for 10000_100 collection. The curve shows the normal distribution that fits the score distributions of the relevant images (Figure 1). The other kernels (not shown) present similar score distributions. Also, we obtained similar results for the 5000_100 collection, in both color spaces, for all queries.

As an observation, the score distributions are different for the different kernel types used in the experiments (Figure 1). While all polynomial kernels and Rad_1, Rad_3, Rad_5 kernels fit the score distribution models given in Section (2.1) (i.e. normal distribution for relevant images and exponential distribution for

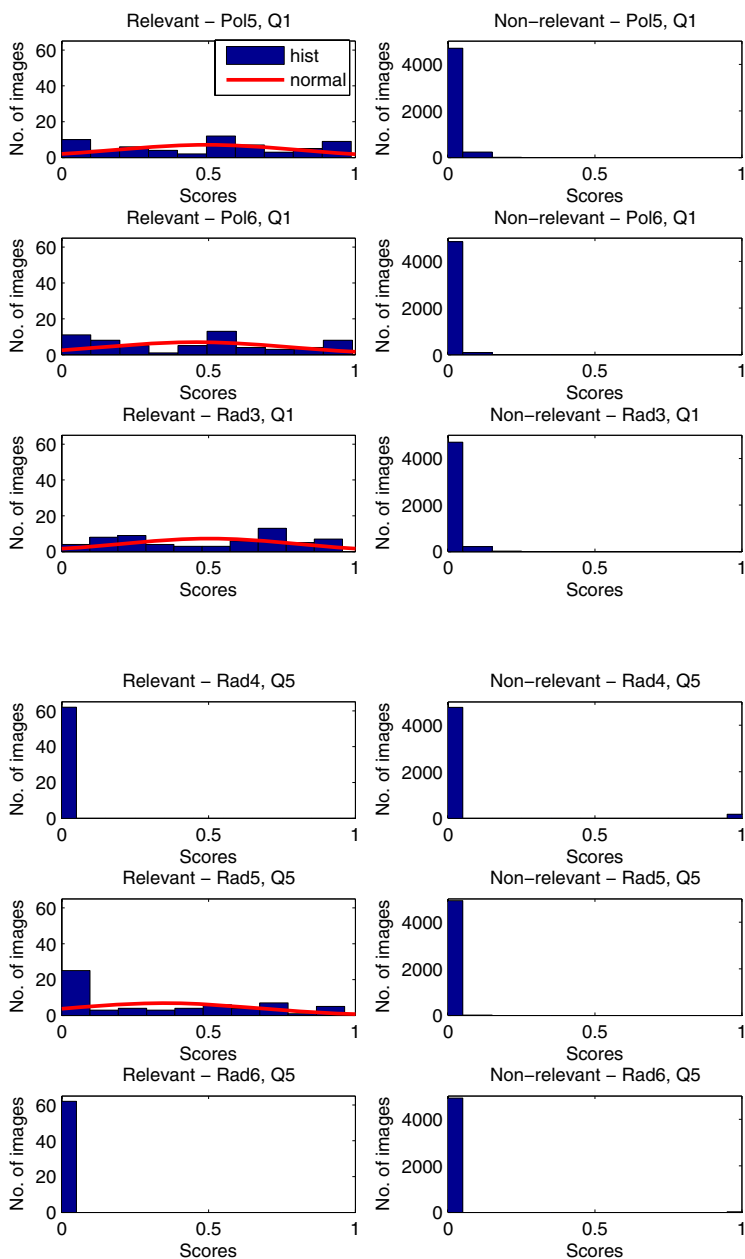


Fig. 1. Score distributions for query Q_1 for 10000_100 in RGB, for $Pol_5, Pol_6, Rad_3, Rad_4, Rad_5, Rad_6$ kernels

Table 2. Parameters used for the different kernels

d	Name	a	b	c	name
$d = 1$	Pol_1	$a = 1$	$b = 2$	$c = 1$	Rad_1
$d = 2$	Pol_2	$a = 1$	$b = 1$	$c = 1$	Rad_2
$d = 3$	Pol_3	$a = 0.5$	$b = 2$	$c = 1$	Rad_3
$d = 4$	Pol_4	$a = 0.5$	$b = 1$	$c = 1$	Rad_4
$d = 5$	Pol_5	$a = 0.25$	$b = 2$	$c = 1$	Rad_5
$d = 6$	Pol_6	$a = 0.25$	$b = 1$	$c = 1$	Rad_6

a) polynomials b) radials

non-relevant images), the other kernels (Rad_2, Rad_4, Rad_6) fit exponential distributions for both relevant and non-relevant images. This is in concordance with the curves these kernels (Rad_2, Rad_4, Rad_6) display in Figure 2, with almost constant R_{norm} values.

Our method of selecting the best kernel tries to fit the score distributions of both relevant and non-relevant images, such that there are as many as possible relevant images with high scores grouped towards the right half of the plot, and less relevant images grouped in the left half side, and vice-versa for the non-relevant images.

For example, in Figure 1, Pol_5 and Pol_6 have very close score distributions, with Pol_6 being slightly better than Pol_5 . When comparing Pol_6 with Rad_3 , Pol_6 shows more scores of the non-relevant images close to 0 than Rad_3 kernel, but this difference is very small. For the relevant images, Rad_3 performs better, i.e. more relevant images get scores higher than 0. Therefore the winner between the two kernels is Rad_3 . A similar discussion can be made to compare Rad_3 and Rad_5 , with the same winner Rad_3 . Interesting, Rad_4 and Rad_6 kernels show many relevant and non-relevant images grouped together (score close to 0).

By using our procedure given in Section 2.3, the kernels are ordered from the best score distribution to the worst score distribution, as follows: $Rad_3, Rad_5, Pol_6, Pol_5, Rad_4, Rad_6$. That is, for query Q_1 the best kernel found by our method is Rad_3 .

Now, further, we want to check how good our kernel selection method performed. For this, we performed extensive experiments on both collections to see the behavior of the different kernels. To evaluate the quality of the retrieval, we use the R_{norm} measure [3]. For each of the image query, R_{norm} compares the expert provided ranking (known) against the system ranking.

For each query, we plot the R_{norm} values (at each feedback step of 10 images) for each kernel. Space limitation precludes us from showing all the plots obtained from our experiments. However, as an example, in Figure 2 we present the plots of the kernel values obtained for five queries $Q_1, \dots, Q_5$ for 10000_100 image test collection in RGB color space.

From Figure 2, one can notice that Rad_5 and Rad_3 kernels show very similar curves, which display significantly higher results than the curves corresponding to the other kernel types. That is, these two kernels are a better choice than

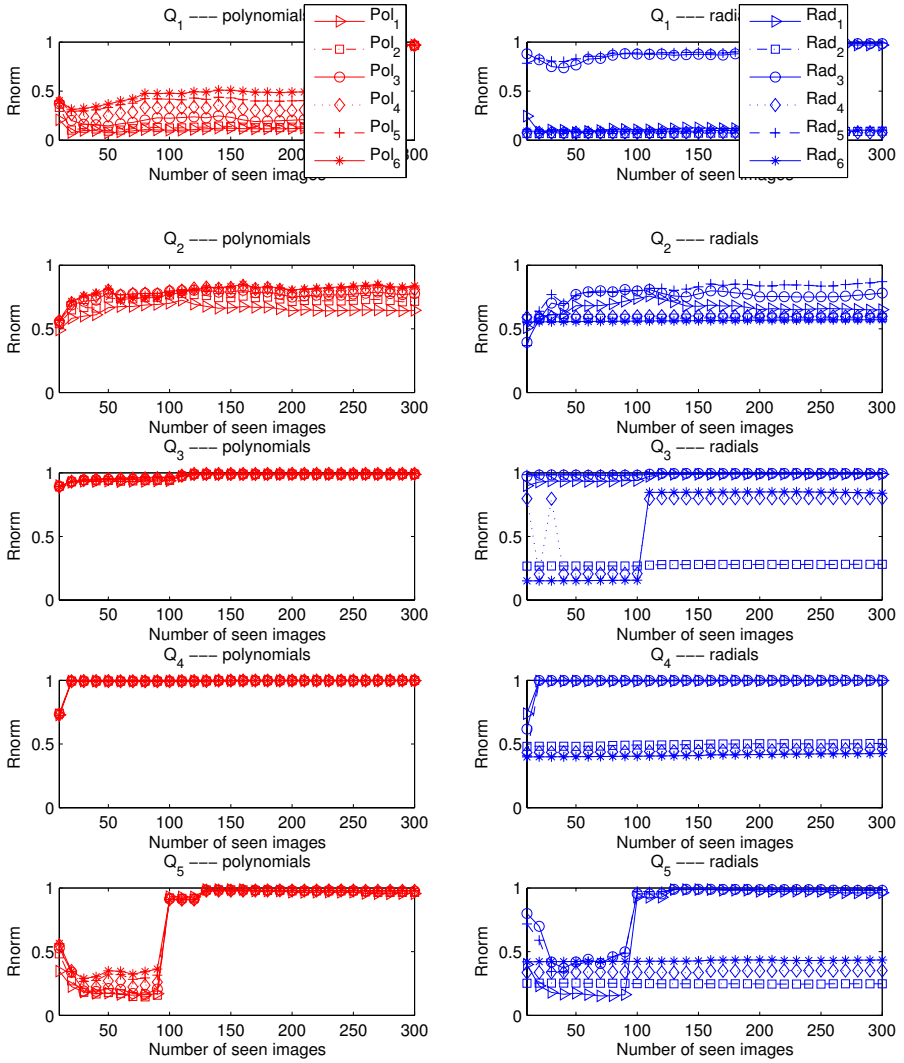


Fig. 2. Kernel results for query Q_1, Q_2, Q_3, Q_4, Q_5 for 10000_100 in RGB

the other kernels. However, if one considers the efficiency issue, then the Rad_3 kernel should be preferred over Rad_5 , since it might reduce the computational cost during retrieval. That is, from our experiments, the retrieval system should use the Rad_3 kernel to achieve best effectiveness and efficiency.

This result is consistent with the result obtained by using our score distribution procedure for selecting the best kernel (same selected kernel type). That is, our method could be a viable solution to automatically select the kernel type in an AIRS.

4 Conclusions and Future Work

Kernel methods offer an elegant solution to increase the computational power of the linear learning algorithms by facilitating learning indirectly in high-dimensional feature spaces. Therefore, kernels are important components that can improve the retrieval system.

Previous work [9,10,8] modeled the score distributions of search engines for relevant and non-relevant documents on a per query basis. Our previous work [6,7] revealed that there is no overall best kernel, but just maybe a best kernel for each query. Based on these works, we propose a kernel selection procedure that automatically chooses the best kernel type for a query by using the score distributions of the relevant and non-relevant images given by user as feedback.

For testing our method, several kernels having polynomial and Gaussian Radial Basis Function (RBF) like forms (6 polynomials and 6 RBFs) are applied to generic images represented by color histograms in RGB and HSV color spaces. We test our method on two image collections of sizes 5000 and 10000 and select the best kernel corresponding to each query. Then, from extensive experimental results on these collections we select the desired kernel to be used for each query. Our method found the same best kernel type for a query as the desired kernel displayed by our extensive experiments.

As future work, we plan to investigate the usage of our kernel selection method with other kernel-based learning methods such as Kernel Perceptron and Kernel SVMs. Another interesting issue to investigate is, maybe, the applicability of the different mixture models of the score distributions to the selection of the best kernel for a particular query in an image retrieval system.

References

1. Doloc-Mihu, A., Raghavan, V., V., Bollmann-Sdorra, P.: Color Retrieval in Vector Space Model. Proceedings of the 26th International ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval, MF/IR (2003)
2. Raghavan, V., V., Wong, S. K. M.: A Critical Analysis of Vector Space Model for Information Retrieval. *Journal of the American Society for Information Science* **37** (1986) 279-287
3. Bollmann-Sdorra, P., Jochum, F., Reiner, U., Weissmann, V., Zuse, H.: The LIVE Project - Retrieval Experiments Based on Evaluation Viewpoints. Proceedings of the 8th International ACM SIGIR Conference on Research and Development in Information Retrieval (1985) 213-214
4. Chapelle, O., Haffner, P., Vapnik, V.: SVMs for Histogram-Based Image Classification. *IEEE Transactions on Neural Networks* **5** (1999) 1055-1064
5. Huijsmans, D. P., Sebe, N.: How to Complete Performance Graphs in Content-Based Image Retrieval: Add Generality and Normalize Scope. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27** (2005) 245-251
6. Doloc-Mihu, A., Raghavan, V., V.: Selecting the Kernel Type for a Web-based Adaptive Image Retrieval System (AIRS). Internet Imaging VII, Proceedings of SPIE-IS&T Electronic Imaging, SPIE **6061** (2006) 60610H

7. Doloc-Mihu, A., Raghavan, V., V.: Using Score Distribution Models to Select the Kernel Type for a Web-based Adaptive Image Retrieval System (AIRS). International Conference on Image and Video Retrieval (CIVR 2006), LNCS, Springer, July 13-15, 2006 (to appear).
8. Swets, J., A.: Information Retrieval Systems. *Science* **141** (1963) 245-250
9. Manmatha, R., Feng, F., Rath, T.: Using Models of Score Distributions in Information Retrieval. Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval (2001) 267-275
10. Arampatzis, A., Beney, J., Koster, C., van der Weide, T., P.: Incrementally, Half-Life, and Threshold Optimization for Adaptive Document Filtering. The 9th Text REtrieval Conference (TREC-9) (2000)
11. Cristianini, N., Shawe-Taylor, J.: An introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press (2000)
12. Tipping, M., E.: The Relevance Vector Machine. *Advances in Neural Information Processing Systems* **12** (2000) 652-658
13. Zhang, Y., Callan, J.: Maximum Likelihood Estimation for Filtering Thresholds. Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval (2001)
14. Smith, J., R.: Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression. PhD thesis, Columbia University, 1997
15. Heckbert, P., S.: Color Image Quantization for Frame Buffer Display. Proceedings of SIGGRAPH **16** (1982) 297-307

Business Intelligence in Large Organizations: Integrating Which Data?

David Maluf¹, David Bell², Naveen Ashish⁴, Peter Putz², and Yuri Gawdiak³

¹ NASA Ames Research Center, Moffett Field CA 94035
David.A.Maluf@nasa.gov

² Universities Space Research Association

³ NASA Headquarters, 300 E SW St, Washington DC 20546

Yuri.O.Gawdiak@nasa.gov
UC-Irvine, 4308 Calit2 Building, Irvine CA 92697
ashish@ics.uci.edu

Abstract. This paper describes a novel approach to business intelligence and program management for large technology enterprises like the U.S. National Aeronautics and Space Administration (NASA). Two key distinctions of the approach are that 1) standard business documents are the user interface, and 2) a “schema-less” XML¹ database enables flexible integration of technology information for use by both humans and machines in a highly dynamic environment. The implementation utilizes patent-pending NASA software called the NASA Program Management Tool (PMT) and its underlying “schema-less” XML database called Netmark. Initial benefits of PMT include elimination of discrepancies between business documents that use the same information and “paperwork reduction” for program and project management in the form of reducing the effort required to understand standard reporting requirements and to comply with those reporting requirements. We project that the underlying approach to business intelligence will enable significant benefits in the timeliness, integrity and depth of business information available to decision makers on all organizational levels.

1 Introduction

The Program Management Tool is a custom-built business intelligence solution for NASA to successfully manage large programs. It enables program and task managers to communicate success critical information on the status and progress of all program levels in an efficient and always current manner. PMT keeps track of project goals, risks, milestones and deliverables, and assists the proper allocation of financial, material, and human resources. It is well integrated with other agency-wide information systems.

PMT is like the nervous system of a complex organization like NASA—it rapidly transports vital information from its operating parts to decision-making authorities and back, allowing a program to move ahead in a coherent and efficient manner and

¹ Extensible Markup Language - <http://www.w3.org/XML/>

to master changes in the environment fast and proactively. PMT supports all essential program management activities and corresponding documents like: creation and monitoring of annual task plans; monthly reporting of technical, schedule, management, and budget status; tracking budget phasing plans; analyzing program risks and mitigation strategies; reporting and evaluating project life cycle costs; accessing convenient aggregated views; automatically creating Earned Value Management assessments, Quad-Charts and other reports. PMT is more of an integrator than a standalone thus connecting multiple distributed resources across the agency; namely ERASMUS reporting system and to the NASA Technology Inventory Database (NTI), and the Integrated Financial Management System (IFMP) thereby significantly reducing cost and time for entering the same data multiple times into different systems. The PMT data architecture is designed to support model-based program management by abstracting NASA 7120.5 project planning guidelines into the PMT either directly via data elements and management monitoring algorithms, and/or via middleware links to portfolio, risk, configuration, and other tools.

Management of large technology programs involves a significant amount of information, which is often delivered using standard templates that meet monthly, quarterly and yearly reporting requirements. Managers who need to be able to gather and understand large amounts of business information in busy meetings have a vital interest in the standardization of report formats. Therefore, information is not only required based on templates, but also required in the format of specific software packages such as Microsoft® Excel, Microsoft® Word, Microsoft® Powerpoint® and Adobe® Acrobat®. While the source of this information is often found in multiple databases with multiple schemas, and the destination of this information often goes to multiple databases with multiple schemas, the human readable format is most critical since management decisions are typically based on conversations and exchanges between humans not machines. Standard formats are required for exchanges between technical experts and decision-making authorities in order for the decision-making authorities to make informed choices using comparable information across the portfolio of selected projects.

Integrated business intelligence is an approach that not only enables automated exchange of information between machines, but also enables automated use of standard business documents, spreadsheets and presentations that are an integral part of the exchange of information between humans for decision making. Business documents become the user interface, and information entered once in business documents gets automatically aggregated and repurposed into standard outputs for use by both humans and machines. This approach is enabled by patent-pending software developed at the U.S. National Aeronautics & Space Administration (NASA), based on an approach with the following key distinctions:

- (1) Standardized business documents are a key user interface (spreadsheets, presentations, documents), eliminating the need for information administration by automating the communication of information using standard templates, and
- (2) A 'schema-less' XML database enables integration and query-based document composition, eliminating the need for database administration by automating the integration of information that have diverse schemas.

A key breakthrough of this approach is that the use of business documents as the inputs and outputs of the management system enables transparency up and down the management chain. Project managers can see how their project information gets summarized in the diversity of presentations and documents used by program managers. Program managers can also see the more detailed project information beyond the project summaries they normally receive. This approach eliminates discrepancies between multiple documents that use the same project information, since the multiple project documents are automatically created from the XML database.

2 Design Requirements

For the management of large technology programs the integration of heterogeneous and distributed technology information is required, and is feasible with modern information technologies. Heterogeneous information involves diverse data structures in structured databases, and unstructured information stored in spreadsheets, word processing documents, presentations and other formats used for human analysis and communication. In the end, it is humans who make management decisions, so this is not just an integration challenge, but also a challenge of composing information from heterogeneous sources for use in human communication and decision-making.

A key to addressing the data heterogeneity issue is the adoption and enforcement of standards across information systems. While XML is clearly emerging to be the *lingua franca* for databases and internet systems, it is encouraging to see the enforcement of standards even at the level of Microsoft® Word and Microsoft® Powerpoint® documents. For instance in the NASA enterprise the new ESR&T² program has explicitly specified the requirements and formats for monthly project reports. Each project is required to submit a monthly progress report, in accordance with its approved project plan or contract Statement of Work, that provides information on the progress of the project during the previous month, in the areas of technical accomplishments, status against schedule, spending, performance against Earned Value Management metrics, planned activities for the following month, and other information pertinent to the tracking and management of the project. The formats have been specified for Word and Powerpoint documents which will be the reporting formats for this program.

The key capabilities that thus must be incorporated in an effective program management information system for an enterprise such as NASA are summarized as follows:

Integrating information from heterogeneous technology databases in a consistent manner, for example project, program and enterprise-level technology databases; past, current, and future technology databases; technical, managerial and financial databases.

Composing analyses and reports from heterogeneous technology databases in support of diverse technology management processes, such as gap analyses, technology assessments, and technology roadmaps; and, historical problem/failure trend reports, bug/requirements tracking, concept/feature tracking.

² Exploration systems research and technology.

Communicating technology information among diverse technology development stakeholders and technology database systems in order to assess data pedigree and support effective decision making for diverse organizational processes, such as project and program management, internal and external reviews, intellectual property and investment management, gap analyses and technology forecasting; and diverse database systems.

3 PMT Features

PMT has been developed to meet the above mentioned requirements and the vision of a business intelligence solution that enables humans to communicate efficiently across organizational boundaries by seamlessly integrating heterogeneous data sources.

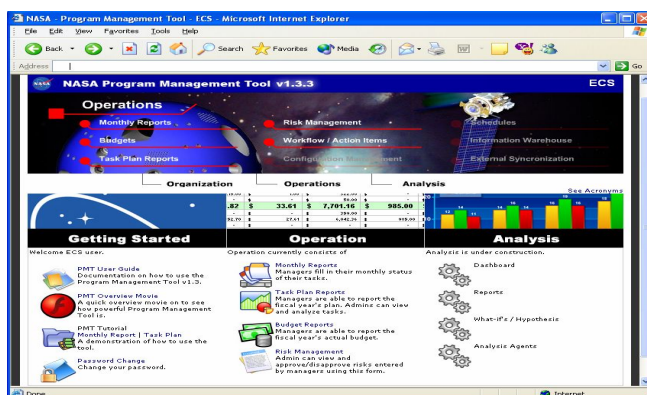


Fig. 1. PMT Portal

The two key distinctions of the approach to integrated business intelligence incorporated in PMT are: 1) standardized business documents as a key user interface, and 2) a “schema-less” XML database enables integration. These key features distinguish PMT from an entire category of existing project and program management tools that are commercially available. The design philosophy of PMT has been to develop a powerful program management information system while providing a simple and intuitive user interaction through standardized business documents in all stages of the program management lifecycle. PMT supports all essential program management activities and corresponding documents such as creation and monitoring of annual task plans; monthly reporting of technical, schedule, management, and budget status; tracking budget phasing plans; analyzing program risks and mitigation strategies; reporting and evaluating project life cycle costs; accessing convenient aggregated views; and automatically creating Quad-Charts and other reports. It is a comprehensive, web-enabled tool which provides an intuitive and enhanced web interface for all user activities from setting up the basic program configuration, to creating reports and

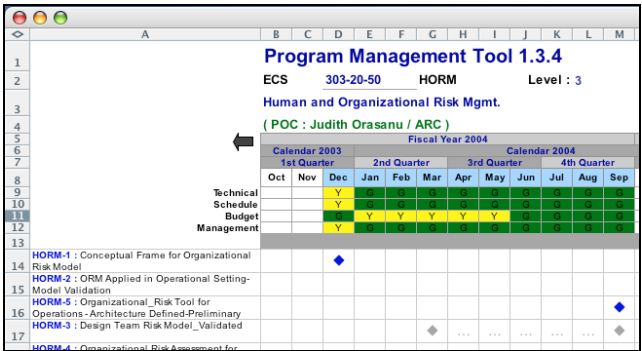


Fig. 2. Spreadsheet as PMT Input

charts. Alternatively, data input can be easily done off-line by downloading annual task plans, monthly reports and other documents to a local desktop. They can be edited with conventional spreadsheet software and uploaded to PMT by the push of a button. Figure 2 provides an illustration of a typical input document:

At NASA input documents such as Task Plans and Monthly reports are typically in formats such as Microsoft® Excel, Microsoft® Word, Microsoft® Powerpoint® and Adobe® Acrobat® since they can be easily distributed by email. PMT users just take the spreadsheets, presentations, and reports that they use in their day-to-day activities to input data into the PMT system. This input process is extremely simple, typically requiring users to upload information through a web server, or simply drag and drop documents into (Web DAV³) folders on the desktop. The input data is automatically converted into XML and stored in a high-throughput information management system (discussed in detail in the Technical Details section).

PMT can automatically generate aggregated documents from several input documents from multiple programs, very quickly. For instance PMT has been used to automatically generate NASA IBPDs (Integrated Budget Performance Documents) spreadsheets using rollup approaches, thus saving significant time and eliminating budget errors. The details and benefits are discussed further in the conclusions section. Other examples of the aggregated documents are integrated budget/cost reports or summary Quad-Charts that can be automatically created by PMT.

PMT is also interfaced and interoperable with existing NASA information systems. For instance it is connected to the ERASMUS⁴ reporting system and to the NASA Technology Inventory⁵, thereby significantly reducing cost and time for entering the same data multiple times into different systems.

In essence, PMT acts like the nervous system of a complex living organism—it rapidly transports vital information from its operating parts to decision-making authorities and back, allowing a program to move ahead in a coherent and efficient manner and to master changes in the environment fast and proactively.

³ Discussed shortly.
⁴ http://www.hq.nasa.gov/erasmus/splash_nasa.htm : A NASA project performance dashboard.
⁵ <http://inventory.gsfc.nasa.gov/> : Database of NASA ongoing technology activities.

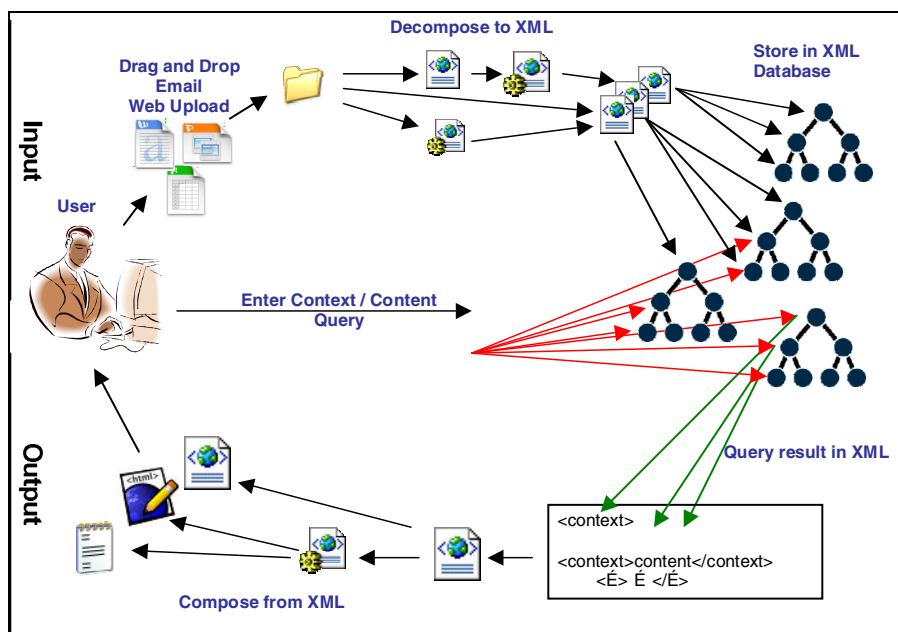


Fig. 3. Document Workflow

A schematic illustration of the information workflow is provided in Figure 3 above. While information storage and other details are discussed in the following section, we must stress that this approach manages information using open and interoperable standards (i.e., XML) but without enforcing day to day users to mark-up their data in any particular format. We begin with enterprise information in business documents which is converted automatically to XML for storage, management, and integration, and which is then composed and formatted for output to business documents and applications as needed for end applications.

4 PMT Architecture

A high level architectural description of PMT is illustrated in Figure 4 on the next page. The architecture is based on two primary levels: 1) client and 2) Netmark-XDB⁶ service; the figure also shows on a more detailed level internal components and communication between components.

At the heart of PMT is a high-throughput information integration and management system, also developed at NASA, called Netmark-XDB [1]. The key design considerations held in the Netmark system design are outlined below:

- Enabling meaningful government-to-government information sharing using international standards, in order to improve mission safety and success

⁶ The term Netmark and XDB are used interchangeably in this paper.

- Enabling effective information management systems that utilize indigenous user interfaces (e.g., spreadsheets as user interfaces), in order to improve system usability
- Enabling rapid development of customized system applications with an enabling platform, in order to reduce time from system concept to system deployment
- Eliminating the need for database analysts using a ‘schema-less’ design, in order to reduce system maintenance costs

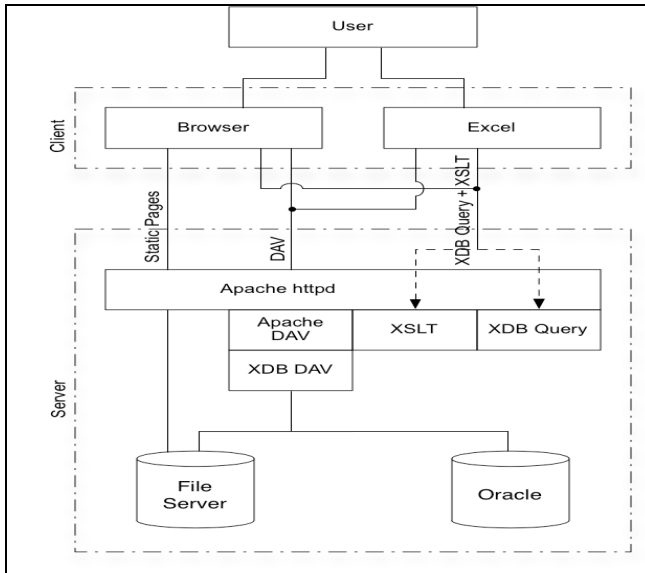


Fig. 4. PMT Software Architecture

Netmark supports the XML standard for metadata and information interchange thus making it an open system that is interoperable with a wide variety of other information systems and tools. Netmark is essentially a data management system for “semi-structured” data i.e., data in the above enumerated kinds of business documents (such as reports in Word or PDF, presentations in formats such Powerpoint, Excel spreadsheets etc.) which does not have a formal “schema”, but where there is indeed some structure in the document implicit from the format and layout. Data loaded into Netmark (done by simple drag and drop) is converted into XML and stored in the data store. The data store is an XML-over-relational data management system. Data can be retrieved very conveniently from Netmark using simple Web-based interfaces that require simple keyword-based input. Netmark also provides powerful data aggregation and composition capabilities.

Clients communicate with the XDB service exclusively over the standards-compliant HTTP⁷ and WebDAV⁸ protocol extensions. The protocol-specific headers,

⁷ HyperText Transport Protocol – <http://www.w3.org/Protocols/>

⁸ Web Distributed Authoring and Versioning Protocol – <http://www.webdav.org/specs/>

client authentication, and feature negotiations are managed by the HTTP server framework—currently Apache's HTTPD⁹. Information storage, retrieval, and recombination modules connect to the Apache server and provide their services through handler API's as defined in the Apache framework.

The XDB DAV module handles WebDAV requests for document storage and retrieval—including decomposition via a pipeline system and storage of decomposed documents in the XDB data store. Information is protected with a full-featured access control mechanism to ensure data security. The XDB Query module provides XDB datastore querying capabilities including content and context querying, information recombination, server-side transformation (using the Xalan¹⁰ XSLT¹¹ processor), and information access control enforcement.

5 Schema-Less Data Integration

The core data management and integration technology in PMT is Netmark, which is designed to keep data-integration middleware 'lean'. Traditional data integration systems, and data management systems in general, are heavily schema-centric [3]. Consequently any enterprise information integration task involves a heavy investment and overhead in managing schemas and further integrating and reconciling schemas (and ontology representations if any). The Netmark approach [2] challenges the paradigm that schema maintenance and merging overhead are a necessary cost; rather in our approach schemas are introduced only as and when necessary with only two relational tables sufficing to represent and store any semi-structured document.

6 Conclusions

PMT has been in use at NASA for about four years, initially for all projects in a \$40M technology research and development program, subsequently for all projects in a \$500M+ research and development program, and has begun to be used by some projects and is under consideration for use by all projects in a multi-billion dollar technology research and development program. PMT has been cited as a key source of the successful passing of the \$40M program's Non-Advocate Review (NAR) by the U.S. Office of Management and Budget, and as a factor in the passing of the \$500M+ program's NAR. The adoption and positive feedback from program manager's and non-advocate reviewers provides qualitative measure of positive results from using PMT.

Quantitative results can be measured in terms of the integrity of the business documents used for management of these technology programs. As one measure of the integrity of information, program and project documents utilized before and after the implementation of PMT on the \$500M+ program were compared, with emphasis on discrepancies that occurred between centrally administered registries and local project

⁹ Apache HTTPD Version 2 – <http://httpd.apache.org/>

¹⁰ Apache Xalan C++ XSLT Engine – <http://xml.apache.org/xalan-c/index.html>

¹¹ The Extensible Stylesheet Language – <http://www.w3.org/Style/XSL/>

Table 1. Differences in milestone data for a major program prior to the use of PMT

	Different MS #		Different Date		Different MS Description		Any Difference	
	Count	%	Count	%	Count	%	Count	%
match	267	85.3%	259	82.7%	232	74.1%	185	59.1%
mismatch	43	13.7%	51	16.3%	74	23.6%	121	38.7%
missing	7	1.0%	7	1.0%	7	2.2%	7	2.2%
Total	313	100.0%	313	100.0%	313	100.0%	313	100.0%

Table 2. Differences in milestone data for specific projects[1-7] in major program prior to use of PMT

		Different MS #		Different Date		Different MS Description		Any Difference?	
		Count	%	Count	%	Count	%	Count	%
1	match	57	76.0%	66	88.0%	54	72.0%	37	49.3%
	mismatch	18	24.0%	9	12.0%	21	28.0%	38	50.7%
	Total	75	100.0%	75	100.0%	75	100.0%	75	100.0%
2	match	38	100.0%	38	100.0%	38	100.0%	38	100.0%
	mismatch	0	0.0%	0	0.0%	0	0.0%	0	0.0%
	Total	38	100.0%	38	100.0%	38	100.0%	38	100.0%
3	match	28	90.3%	29	93.5%	27	87.1%	26	83.9%
	mismatch	2	6.5%	1	3.2%	3	9.7%	4	12.9%
	missing	1	3.2%	1	3.2%	1	3.2%	1	3.2%
	Total	31	100.0%	31	100.0%	31	100.0%	31	100.0%
4	match	46	90.2%	45	88.2%	35	68.6%	29	56.9%
	mismatch	5	9.8%	6	11.8%	16	31.4%	22	43.1%
	missing	0	0.0%	0	0.0%	0	0.0%	0	0.0%
	Total	51	100.0%	51	100.0%	51	100.0%	51	100.0%
5	match	21	100.0%	20	95.2%	18	85.7%	17	81.0%
	mismatch	0	0.0%	1	4.8%	3	14.3%	4	19.0%
	missing	0	0.0%	0	0.0%	0	0.0%	0	0.0%
	Total	21	100.0%	21	100.0%	21	100.0%	21	100.0%
6	match	15	42.9%	9	42.9%	9	42.9%	5	23.8%
	mismatch	4	47.6%	10	47.6%	10	47.6%	14	66.7%
	missing	2	9.5%	2	9.5%	2	9.5%	2	9.5%
	Total	21	100.0%	21	100.0%	21	100.0%	21	100.0%
7	match	62	81.6%	52	68.4%	51	67.1%	33	43.4%
	mismatch	10	13.2%	20	26.3%	21	27.6%	39	51.3%
	missing	4	5.3%	4	5.3%	4	5.3%	4	5.3%
	Total	76	100.0%	76	100.0%	76	100.0%	76	100.0%

documents around milestones and deliverables. Milestones and deliverables represent one of the most important aspects of the agreement between program and project managers. Differences in program and project manager’s understandings of the agreed upon milestones can have significant impact on the project’s and program’s success.

Table 1 shows that prior to the use of PMT only under 60% of all level three project milestones had complete integrity between the project's documents and the program's documents. Table 2 shows the breakdown in information integrity for each of seven major projects, with levels of information integrity varying from under 24% to 100%. With the use of PMT the integrity of business documents became 100% throughout, output documents were automatically created from input documents without any manual cutting & pasting of information.

References

- [1] David Maluf and Peter Tran, "NETMARK: A Schema-less Extension for Relational Databases for Managing Semi-Structured Data Dynamically." Proceedings of the 14th International Symposium on Methodologies for Intelligent Systems, October 28-31, 2003, 231-241
- [2] Naveen Ashish, David Bell, Chris Knight, David Maluf and Peter Tran: "Semi-structured Data Management in the Enterprise: A Nimble, High-throughput and Scalable Approach". International Databases and Engineering Applications Symposium, Montreal, July 2005
- [3] Alon Y. Halevy, Naveen Ashish, Dina Bitton, Michael J. Carey, Denise Draper, Jeff Pollock, Arnon Rosenthal, Vishal Sikka: "Enterprise information integration: successes, challenges and controversies." SIGMOD Conference 2005: 778-787

Intelligent Methodologies for Scientific Conference Management

Marenglen Biba, Stefano Ferilli, Nicola Di Mauro, and Teresa M.A. Basile

Dipartimento di Informatica, Università degli Studi di Bari,
via E. Orabona 4, 70126 Bari, Italia
{biba, ferilli, ndm, basile}@di.uniba.it

Abstract. This paper presents the advantage that knowledge-intensive activities, such as Scientific Conference Management, can take by the exploitation of expert components in the key tasks. Typically, in this domain the task of scheduling the activities and resources or the assignment of reviewers to papers is performed manually leading therefore to time-consuming procedures with high degree of inconsistency due to many parameters and constraints to be considered. The proposed systems, evaluated on real conference datasets, show good results compared to manual scheduling and assignment, in terms of both accuracy and reduction of runtime.

1 Introduction

Managing scientific conferences involves many complex tasks regarding the organization and scheduling of the resources involved. Setting up a large conference is often quite a hard task because of the many subtasks involved. This process begins with the conference creation from the Program Committee Chair (PCC for short) who provides essential elements for the conference such as: title, topics, place, start date and deadline of papers submission, start date and deadline of papers review, date of main event etc. In particular, an important activity is the selection of Program Committee Members (PCMs). After this phase the *Call for papers* is issued and the papers submission has to be managed. Then, the next task is assigning each paper to the proper reviewers. During the review process the PCMs evaluate each paper assigned to them and produce their assessment. According to such evaluations the papers are selected and finally approved, which yields an input to the next task for the local organization of the conference. Finally, the authors of accepted papers may submit a new version of their paper, in the so-called camera-ready format, to the PCC, who will send them, together with the preface and the table of contents of the book, to the publisher in order to have the proceedings printed.

Thus, Web-based applications have been developed in order to make easier some of the jobs to be carried out. Typically, these systems provide support for: submission of abstracts and papers by Authors, download of papers by the Program Committee (PC), submission of reviews by the PCMs, handling of reviewers preferences and bidding, Web-based assignment of papers to PCMs for review, review progress tracking, Web-based PC meeting, notification of acceptance/rejection, sending notification

e-mails. However, the more knowledge-intensive tasks require a more “clever” support than just a comfortable interface and improved interaction. Two of the hardest and most time-consuming activities are the assignment of reviewers to submitted papers and the sessions' organization and scheduling (commonly said *Timetabling*). This phase concerns the distribution of the submitted papers into a set of sessions and the assignment of the sessions to available rooms and periods of time. Due to the many constraints to be satisfied, performing manually such tasks is very difficult and usually does not guarantee optimal results.

This work aims at showing how the use of intelligent methodologies can bring important improvements in these two activities, by embedding expert components in a more general framework of a scientific Conference Management System. Section 2 introduces the main features of DOMINUS (DOCUMENT Management INtelligent Universal System), a prototype of a general Conference Management System (CMS), and describes how the expert systems are embedded in it. Section 3 describes the review assignment component, while Section 4 deals with the session scheduling one. Finally, Section 5 concludes the paper, with remarks and future work proposals.

2 DOMINUS for Conference

DOMINUS for Conference is a prototype of CMS. It is still under development. Among other basic services, a CMS can be seen as a system collecting documents (submitted) in electronic form. The distinguishing feature of DOMINUS for Conference lies in its ability to understand the semantics of the document components and content, and to exploit such an understanding to support conference organizers in carrying out their most difficult, knowledge-intensive and time-consuming activities. DOMINUS for Conference is the tailoring to scientific conference management of the general-purpose system DOMINUS for intelligent document analysis, processing and management [7].

DOMINUS implements artificial intelligence methods and techniques for document analysis and understanding. It performs layout analysis on electronic documents in order to exploit the components identified in the document structure to understand their significance and extract from them relevant information about the document. The final aim is to build, through the use of machine learning methods, intelligent semantic-based search engines that offer access to information based on semantic contents of analyzed documents. The entire process of document processing passes through several phases: Document Layout Analysis, that identifies the geometric component in terms of blocks of the document; Document Image Classification that identifies the layout class the document belongs to; Document Image Understanding that identifies the semantic role (according to the document class) of the various layout component identified in the first phase and labels them accordingly; Text Extraction that reads and records the text inside the significant components identified and labeled in the previous phase; Text Categorization during which the text extracted is used to generate key words that describe the topic the document is about; and lastly Information extraction that aims at exploiting all the relevant results of the previous phases to extract and structure significant and important information about the document content.

Once an author has submitted his own paper in an electronic format, it undergoes the processing steps described above. A first exploitation of automatic document processing is that the text, extracted as a result of the six phases above, can be used to automatically check the paper compliance to the conference standards (e.g., layout style and page number) and file the submission record, without requiring the author to manually fill the submission fields (title, authors, affiliations, abstract, number of pages, topics). Then, some of these data (e.g., authors, affiliation and topics) can be exploited by the expert system GRAPE (Global Review Assignment Processing Engine) as an input to carry out the assignment of reviewers to papers. Also the reviewer expertise topics can be automatically inferred by processing their CVs and a selection of their papers: indeed, experience proves that topics manually entered by the reviewers sometimes do not completely represent their actual research interests according to their publications. Lastly, information on paper authors, title and topics can be exploited also by the expert system for session scheduling, in order to carry out its task.

Thus, DOMINUS for Conference aims at defining a multifunction framework for conference management activities that spreads from basic browsing services to scheduling sessions and presentations leading therefore to a complete solution for conference handling based entirely on intelligent components. Its Web version helps organizers of conferences in managing efficiently all their activities through Web accessibility in order to provide a valid tool for conference organizing committees.

3 The GRAPE System

The review process starts after the deadline for paper submission. Then, suitable PCMs are selected, which will act as reviewers, in order to evaluate the submitted papers. Therefore, the PCC sends to individual reviewers the collected submissions with review forms consisting of a set of questions to assess the quality of the paper, that the Reviewers must fill in and return to the PCC. Each submission is typically examined and evaluated by 2 or 3 reviewers. Generally, the review process ends with the PC meeting, where papers' rejection or acceptance for presentation at the conference are discussed on the basis of collected reviews. After this meeting, extracts of the reviewers' comments are typically sent back to all the authors, so that they can improve their paper.

In the current practice, before the submission phase starts, the Chair usually sets up a list of research topics of interest for the conference and asks each reviewer to specify which of them correspond to their main areas of expertise. Then, during the submission process, authors are asked to explicitly state which topics apply to their papers. Such an information provides a first guideline for associating reviewers to papers. However the presence of many constraints makes the problem a very hard one to cope with in terms of time of assignment, for a number of reasons. For instance, some reviewers cannot evaluate specific papers due to conflicts of interest or to their little expertise in the paper topics, or because they are not willing to revise more than a certain number of papers.

GRAPE [1] is an expert system, written in CLIPS (C Language Integrated Production System), that performs the reviewers assignment by taking advantage from both the papers content (topics) and the reviewers preferences (biddings). It can be set to

exploit the papers topics only, or both the paper topics and the reviewers' biddings. Its fundamental assumption is to prefer the topics matching approach over the reviewers' biddings one, based on the idea that they give to assignments more reliability. Then, reviewers' preferences can be used for tuning the paper assignments: they are very useful because of the unpredictability of the distribution of reviewers and papers over the list of topics, which causes situations in which some papers have a lot of experts willing to review them, while some others simply do not have enough reviewers.

Let $\{p_i\}_{i=1,\dots,n}$ denote the set of papers submitted to the conference, t be the number of conference topics, and $\{r_j\}_{j=1,\dots,m}$ be the set of reviewers. The goal is to assign the papers to reviewers, such that: 1) each paper is assigned to exactly k reviewers (usually, 3 or 4); 2) each reviewer has roughly the same number of papers to review (on average nk/m , but additional constraints can be set to indicate that some reviewer can review at most a given number of papers); 3) papers are reviewed by domain experts; and, 4) reviewers revise articles based on their expertise and preferences.

Two measures were defined to guide the system during the search for the best solutions. The *reviewer's gratification* degree of a reviewer is based on: a) the confidence degree between him and the assigned articles: the confidence degree between a paper and a reviewer is equal to the number of topics in common; and b) the number of assigned papers that the reviewer chose to revise (discussed in more detail in Section 4.2). The *article's coverage* represents the coverage degree of an article after the assignments. It is based on: a) the confidence degree between the article and the assigned reviewers (the same as before); and b) the expertise degree of the assigned reviewers, represented by the number of topics in common. The expertise level of a reviewer is equal to the ratio of topics he is expert in on the whole set of conference topics. GRAPE tries to maximize both reviewer gratification and article coverage degree during the assignment process, in order to fulfill requirements 3) and 4). To reach this goal it is mandatory that each reviewer provides at least one topic of preference, otherwise the article coverage degree would be null.

The two main inputs to the system are the aforementioned sets of papers and reviewers. Each paper is described by its title, author(s), affiliation(s) of the author(s) and topics TP_i . On the other hand, each reviewer is described by his name, affiliation and topics of interest Tr_j . Furthermore, the system can take as input a set of constraints indicating (i) the possibly specified maximum number of reviews per Reviewer and (ii) the papers that must be reviewed by a reviewer because of specific indication by the PCC. It can be also provided with a set of conflicts indicating which reviewers cannot evaluate specific papers under suggestion of the PCC: in any case, such a set is enriched by GRAPE by deducting additional conflicts between papers p_i and reviewers r_j whenever r_j is a (co-)author of p_i , or the affiliation of r_j is among the affiliation(s) reported in p_i .

We say that a reviewer can revise a paper with a degree that is equal to the confidence degree between them if it is not null, or else is equal to the expertise degree of the reviewer – ranging in $[0,1]$ – when the confidence degree is 0. Given a paper, the number of candidate reviewers for it is the number of reviewers that can revise it with degree greater than or equal to 1.

3.1 The Assignment Process

The assignment process is carried out in two phases. In the former, the system progressively assigns reviewers to papers with the lowest number of candidate reviewers, thus assuring to assign a reviewer to papers with only one candidate reviewer. At the same time, the system prefers assigning papers to reviewers with few assignments, to avoid having reviewers with zero or few assigned papers. Hence, this phase can be viewed as a search for review assignments by keeping low the average number of reviews per reviewer and maximizing the coverage degree of the papers. In the latter phase, the remaining assignments are chosen by considering first the confidence level and then the expertise level of the reviewers. In particular, given a paper which has not been assigned k reviewers yet, GRAPE tries to assign it a reviewer with a high confidence level between them. In case it is not possible, it assigns a reviewer with a high level of expertise.

The assignments resulting from the base process are presented to each reviewer, that receives the list A of the h assigned papers, followed by the list A' of the remaining ones (both, A and A' are sorted by article's coverage degree). The papers are presented to the reviewer as virtually bid: the first $h/2$ papers of A are tagged "high" (he would like to review), the next h papers are tagged "medium" (he feels competent to review) and all the others are tagged "low" (he does not want / does not feel competent to review). Now, the reviewer can actually bid the papers by changing their tag, but preserving the number of high and medium ones. Furthermore, he can tag $h/2$ papers as "no". All the others are assumed to be bid as low. Only papers actually bid by reviewers generate a preference constraint. When all the reviewers have bid their papers, GRAPE searches for a new solution that takes into account these biddings. In particular, it tries to change previous assignments in order to maximize both article's coverage and reviewer's gratification. By taking the article's coverage high, GRAPE tries to assign the same number of papers bid with the same class to each reviewer. Then, the solution is presented to the PCC. It is important to say that, if the PCC does not like the solution, he can change some assignments and force the system to give another solution satisfying these new constraints as well. In particular, he may: assign a reviewer to a different paper; assign a paper to a different reviewer; remove a paper assignment; or remove a reviewer assignment. The main advantage of GRAPE relies in the fact that it is a rule-based system. Hence, it is very easy to add new rules in order to change/improve its behavior, and it is possible to describe background knowledge, such as further constraints or conflicts, in a natural way. For example, one could add a rule that expresses the preference to assign a reviewer to the articles in which he is cited.

3.2 Evaluation

The system has been evaluated on real-world datasets built by using data from previous European and International conferences. In order to have an insight on the quality of the results, in the following we present some interesting characteristics of the assignments suggested by GRAPE.

For a previous European Conference the experiment consisted in a set of 383 papers to be distributed among 72 Reviewers, with $k = 3$ reviews per paper. The system

was able to correctly assign 3 reviewers to each paper in 152 seconds. Obtaining a manual solution took about 10 hours of manual work from the 4 Program Chairs of that conference. Each reviewer was assigned 14.93 papers on average by topic (when there was confidence degree greater than 1 between the reviewer and the paper), and only 1.03 papers on average by expertise degree (which is a very encouraging result). GRAPE made many good assignments: in particular, it assigned to 55 reviewers all 16 papers by topics.

Another dataset was obtained from the IEA/AIE 2005 conference. It consisted of a set of 266 papers to be distributed among 60 Reviewers. The conference covered 34 topics. In solving the problem, the system was able to correctly assign 2 reviewers to each paper in 79.89 seconds. The system was also able to assign papers to reviewers by considering the topics only (it never assigned a paper by expertise). In particular, it assigned 10 papers to 38 reviewers, 9 to 4 reviewers, 8 to 6 reviewers, 7 to 1 reviewer, 6 to 9 reviewers, 5 to 1 reviewer, and 2 to 1 reviewer, by considering specific constraints for some reviewers that explicitly requested to revise few papers.

As a conclusion, the rule-based system GRAPE, purposely designed to solve the problem of reviewer assignments for scientific conference management, having been tested on real-world conference dataset, revealed how the use of intelligent methodologies based on expert systems helped in improving significantly the entire assignment process in terms of quality and user-satisfaction of the assignments, and of reduction in execution time with respect to that taken by humans to perform the same process.

4 The Resource Scheduling Expert System

Scheduling activities under rather complex constraints have been subject of much research effort. Indeed, a large number of variants of the problem have been proposed in the literature, that differ from each other because of the type of institution involved and the type of constraints. This problem, that has been traditionally considered in the operational research field, has recently been tackled with techniques belonging also to artificial intelligence (e.g. genetic algorithms, tabu search, simulated annealing, and constraint satisfaction) [2], [3], [4], [5]. The intrinsic complexity of the problem makes it important enough to consider specific knowledge-based solutions in order to provide valid tools that could face the problem bringing significant improvements with respect to manual approaches. Since in most organizations the problem of scheduling is quite overwhelming there are really strong motivations to consider applying artificial intelligence methodologies in order to build expert systems that should deal with the task of allocating resources to activities. In many domains assigning available resources to concurrent activities is critical in terms of importance for the organization processes. Because of the many constraints to be satisfied and many parameters to be considered, manual procedures often yield low quality results even in spite of long execution time spent for the entire process of scheduling.

In the current practice all the activities necessary for the generation of a Timetable are done manually. These activities must deal with the satisfaction of rather complex constraints, such as: unavailability of an author, of the Chair of a session or the unavailability of places necessary for the sessions; pre-assignments that impose certain

meetings to be kept at a precise time and/or place; preferences for places or time of a presentation; limitation on place and time, when some places or periods of time are not available for a certain presentation or session; limitation on the sequence and alignment of the sessions when two sessions dealing with the same topic cannot be scheduled to run simultaneously; limitation on the sequence of sessions when the same author cannot attend two simultaneous sessions; reasonable timetable in order to satisfy certain requirements like the maximal number of sessions during a day or the necessity of breaks during lunch time.

4.1 Scheduling in Scientific Conference Organization

One of the activities that involves planning and timetabling is the organization of scientific conferences. The phase of sessions organization requires much effort because of the many parameters to consider. In particular this process consists in the distribution of the papers accepted into sessions and for each session the allocation of available time slices and vacant accommodations. This scheduling activity is heavily constrained by the following variables to be taken into account [6]: Persons: each participant has his own role in the conference thus the scheduling procedure should resolve cases of multiple roles by a person. For instance, an author of a presentation could be Chair of another session, therefore the two sessions cannot be scheduled to run simultaneously; Discussions: these are defined as being any kind of interaction inside the conference like the presentation of a publication, a workshop about a particular topic etc. The constraint the discussions impose on the scheduling process concerns their distribution and organization in non-simultaneous sessions with the same topic and their uniform grouping into session inside which they share the same topics; Sessions: these are the main entities to elaborate in order to produce an optimal timetabling. Constraints directly related to sessions regard the full slice of time necessary for the entire session, the use of specific equipment for the sessions presentations that could be required by another simultaneous session, the presence of not available time periods dedicated to other activities such as lunch breaks which should be taken into account and finally the available accommodations to be shared among concurrent sessions; Timeblocks: in this case the constraints to satisfy are that all the entire time available must be divided in timeblocks which should each contain a session and all the timeblocks should be optimally combined in order to maximally exploit remaining time slices for other sessions; Accommodations: their constraints are related to space requirements of specific sessions, for instance some sessions could be too rich in terms of contents and therefore in terms of participation and need large rooms to proceed. Other sessions could use only particular accommodations because of the particular equipment the rooms provide. Furthermore no accommodations should be shared by different sessions during the same timeblock.

In addition to the constraints introduced above there are different kinds of sessions which means that other parameters must be taken into account. Some of these are: Plenary Sessions that aim at synthesizing the state of the art of a certain research field and that usually run without overlapping with other sessions; Special Sessions that are organized by a person designated on purpose by the Program Committee. These sessions render unavailable the chosen person for other sessions or activities; Sessions of Free Contribution are usually organized by the Program Committee and concern

topics proposed by authors; Poster Sessions are organized to allow the authors to present their publications in particular sessions set up on purpose and that differ from the classic session format.

4.2 The Expert System

In order to provide a solid solution for the scheduling of the activities in scientific conference management contexts that could manage to resolve all the constraints presented in the previous paragraph, an expert system was developed and tested in this domain. The previous approaches in resolving this kind of problem have always referred to other solutions already built for other contexts such as university and school timetabling, enterprise activities scheduling, without considering the possibility to design a tailored solution for the context of conference management.

The expert system built is a rule-based system developed in CLIPS. For the construction of the knowledge base of the system there were formulated 11 functions and 30 rules.

Some of these functions were designed to verify the availability of certain resources for a certain session while others verify if the constraints are still satisfied after changing certain parameters. The controls are performed upon all the entities that constitute the scheduling process. We mention here some of the verification tasks the functions implement: one function verifies if the equipment required for a certain presentation is provided by the accommodation to be assigned to the presentation; another function verifies if an author of a presentation who has expressed his unavailability for a certain period can be available for the time that is to be assigned to his presentation; one function verifies if the author of a presentation which is to be assigned a time slice could have to discuss at the same time another presentation in case of more than one paper submitted; another function verifies the parallelism of the session to which is being assigned a certain presentation and other sessions scheduled to be run at the same time; one function calculates the total time of a session.

The rules encapsulated all the human expertise of the scheduling process trying to encode all the peculiarities of such a process. Some of these rules implement actions such as: help the user indicate the accommodation and the time to be inserted for a presentation that must not be parallelized with any other presentation; find the hours available in a room in a certain period of time; eliminate main sessions for which do not exist presentations; break into the availability of an accommodation distinguishing among morning and afternoon hours; change session being based on compatibility between presentation and session topics in case the main session a presentation belongs to, has as estimated time less than one hour and there are not principal session with the same topic; assign a mobile resource to a presentation that requests the resource guaranteeing the fulfilment of all the constraints; resolve the problem that arises when the available remaining time in an accommodation is smaller than the necessary quantity of time to allocate a certain presentation. In this case the rule resolves the problem searching for another presentation on the same topic in the same room which has been assigned a time slice equal to that needed to end the session time.

In order to help the end user to easily manage the process of session management, a stand-alone application was developed in the programming language Java. It offers

various visual tools with the purpose of providing to the user user-friendly windows for creating new timetables, modify existing timetables, insert new data, visualize existing timetables, cancel existing timetables, update timetables and visualize details. Through the inserting of new data the user could remove or add new constraints about a presentation or add new data like another accommodation, author or a presentation. Once these data were entered, the expert system running in the background is involved and a new timetable is generated taking into account the new data inserted.

From an architectural point of view the application has 3 levels: presentation level which permits to the end-user to interact with the system through a number of views generated by the system; domain level which implements all the domain-specific functionalities in order to support the interaction with the knowledge base; data source level which is responsible for accessing the knowledge base and is the only component to know the logic and physical structure of the knowledge base.

4.3 Evaluation

The system described in the previous paragraph was tested upon a dataset of a real-world conference, IEA/AIE 2005. The presentations and the person who presented them were 127. Before running the system some basic information was entered like the presentations that could not run simultaneously or eventual breaks between sessions. The execution time of the expert system for building a configuration that satisfied all the constraints was about 4 hours which seems a very enthusiastic result compared to the 7 working days of 4 persons that were fully occupied for creating the timetable. The assignment generated by the expert system maintained a very good level of quality. Through the assignment rules it was possible to allocate automatically about 90% of the presentations. The remaining 10 % was managed using the rules that deal with the incompatibilities and that manage to modify the topic of a presentation based on the assignments made previously.

In addition to the good results in terms of quality and time for the specific task, one should also consider the typical advantages coming from the use of expert systems, such as the exploitation of heuristics to guide the search for a solution, the ability to explain the rationale underlying the response and the possibility of straightforwardly extending the knowledge base with new information, even when hundreds of rules are used.

5 Conclusions and Future Work

Organizing scientific conferences involves many complex tasks, whose management requires a more “clever” support than just a comfortable interface and improved interaction. This work showed how the use of intelligent methodologies can bring important improvements in such a domain. Indeed, the scientific Conference Management System DOMINUS for Conference exploits Machine Learning techniques and Expert Systems technology to automatically process submitted documents and effectively/efficiently support conference organizers in two of the hardest and most time-consuming activities: the assignment of reviewers to submitted papers and the sessions' organization and scheduling.

Although results on real-world datasets are encouraging, future work will concern extension and improvement of the system, both as regards its document processing features and as concerns the range of activities supported given by expert components.

References

1. N. Di Mauro, T.M.A. Basile, and S. Ferilli. GRAPE: An Expert Review Assignment Component for Scientific Conference Management Systems. In F. Esposito and M. Ali, editors, *Innovations in Applied Artificial Intelligence (IEA/AIE05)*, volume 3533, LNAI, pp. 789–798, Springer Verlag, 2005.
2. Schaerf A. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87-127, 1999
3. Burke E. K. and Petrovic S. Recent research directions in automated timetabling. *European Journal of Operational Research*, 140: 266--280, 2002.
4. Carter M.W. A Survey of Practical Applications of Examination Timetabling. *Operations Research* 34:193-202. 1986.
5. de Werra D. An introduction to timetabling, *European Journal of Operations Research*, 19:151-162, 1985.
6. Sampson S. E. Practical Implications of Preference-Based Conference Scheduling, *Production and Operations Management*, 13 (3), pp. 205-215, Production and Operations Management Society, Baltimore, October, 2004
7. Esposito F., Ferilli S., Basile T.M.A., Di Mauro N. Semantic-Based Access to Digital Document Databases. In M.-S. Hacid, Z. W. Ras, and S. Tsumoto, editors, *Foundations of Intelligent Systems (ISMIS05)*, volume 3488, LNAI, pp. 373–381, Springer Verlag, 2005.

The Consistent Data Replication Service for Distributed Computing Environments^{*}

Jaechun No¹, Chang Won Park², and Sung Soon Park³

¹ Dept. of Computer Software
College of Electronics and Information Engineering
Sejong University, Seoul, Korea

² Intelligent IT System Research Center
Korea Electronics Technology Institute
Bundang-gu, Seongnam-si, Korea

³ Dept. of Computer Science & Engineering
College of Science and Engineering
Anyang University, Anyang, Korea

Abstract. This paper describes a data replication service for large-scale, data intensive applications whose results can be shared among geographically distributed scientists. We introduce two kinds of data replication techniques, called owner-initiated data replication and client-initiated data replication, that are developed to support data replica consistency without requiring any special file system-level locking functions. Also, we present performance results on Linux clusters located at Sejong University.

1 Introduction

Supporting a consistent data replication mechanism is a critical issue in distributed computing where a large amount of data sets generated from large-scale, data-intensive scientific experiments and simulations are frequently shared among geographically distributed scientists [1]. In such an environment, the data replication technique significantly affects the performance by minimizing the remote data access time.

The usual way of managing consistent data replicas between distributed sites is to periodically update the remotely located data replicas, like implemented in Globus toolkit [2]. This method is implemented under the assumption that the data being replicated is read-only so that once it has been generated would not it be modified in any grid site. This assumption is no longer true in such a case that a remote site may modify or update the data replicated and stored in its local storage. If another remote site tries to use the same data replicated on its storage before the data is updated with the new one, the data consistency between distributed sites would then be failed and the scientist gets the wrong results.

^{*} This work was supported in part by a Seoul R&BD program.

In order to solve this problem by providing the data replication consistency, we have developed two kinds of data replication techniques, called owner-initiated data replication and client-initiated data replication. In the owner-initiated data replication, the data owner who owns the application data sets starts the data replication to share the data sets with remote clients. In the client-initiated data replication, the client who needs the data sets starts the data replication by connecting to the data owner. Furthermore, our data replication techniques do not need to use file system-level locking functions so that they can easily be ported to any file systems.

2 Replication Architecture

2.1 Client Grouping

Our replication architecture supports the client grouping to eliminate the unnecessary communication overheads. By making the client groups, our architecture can easily detect the other clients who share the same data replicas and can let them take the new copy of the modified data, without affecting other clients.

Figure 1 shows the client grouping of our replication architecture. In order to store the metadata information about the client grouping and replication mechanism, six data base tables.

The metadata database tables and the application data sets generated by users are located at the data owner. The remote clients who want to share the application data sets with the data owner are grouped into several client groups, according to the data sets replicated on the local storage. Figure 1 shows two remote client groups created based on the data replicas. Each client in a group

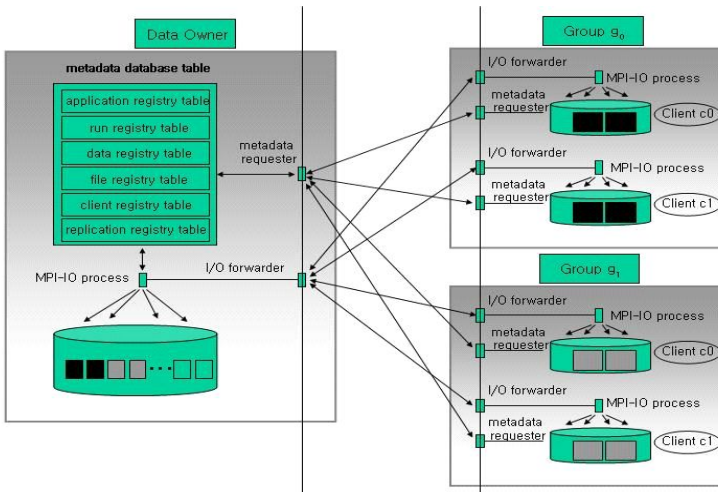


Fig. 1. Client grouping

is identified with groupID and clientID, such as (g_0, c_0) for the first client and (g_0, c_1) for the second client in Group g_0 .

2.2 Owner-Initiated Data Replication

In the owner-initiated data replication, when user generates data sets at the data owner, our architecture replicates them to the remote clients to share the data sets with the data owner. Also, when a remote client changes the data replicas stored in its local storage, it broadcasts the modifications to the members in the same group and to the data owner for replica consistency. Figure 2 shows the steps taken in the owner-initiated replication.

At time t_i , since the client A and client B want to receive the same data sets, a and b , from the data owner, they are grouped into the same client group. When an application generates the data sets at time t_j , a and b are replicated to the client A and client B.

Suppose that the client A modifies the data replicas at time t_k . The client A requests for the IP address of other members in the same group, sends the modified data sets to them, and waits for the acknowledgement.

When the data owner receives the modified data, it updates them to the local storage and sets the status field of the replication_registry_table to "holding" to prevent another client from accessing the data sets while being updated. When the data owner receives the notification signal from the client who initiated the data modification, it sets the status field to "done", allowing another client to use the data replica.

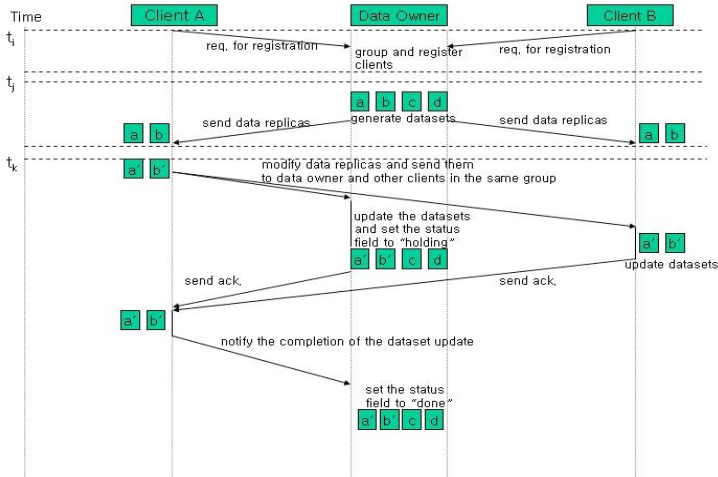


Fig. 2. Owner-initiated replication

2.3 Client-Initiated Data Replication

Figure 3 shows the client-initiated data replication where only when the modified data replicas are needed by users are those data replicas sent to the requesting client and to the data owner. Unlike in the owner-initiated data replication, there is no data communication when users on the data owner produce the application data sets. If a client needs to access the remote data sets stored in the data owner, he will then get the data replica while registering to our architecture.

In Figure 3, after the application generates the data sets at time t_i , the client A and client B request for the data sets, a and b , to the data owner at time t_j . Because they want to receive the same data sets, they become the member of the same client group.

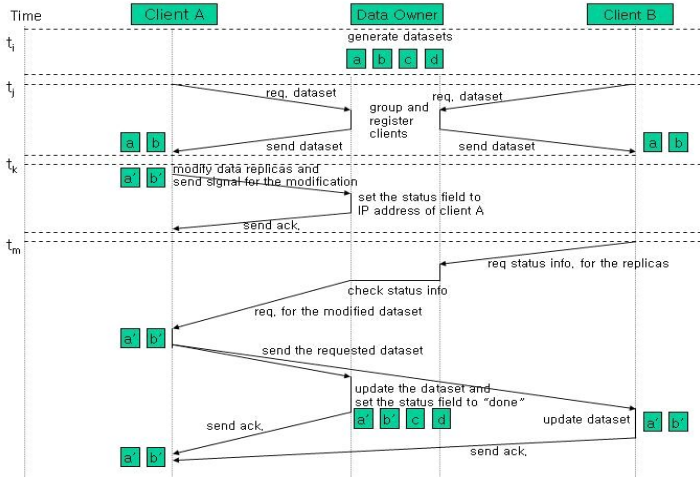


Fig. 3. Client-initiated replication

Suppose that client A modifies the data replica at time t_k . He just sends a signal to the data owner to update the corresponding status field of the data set with the IP address of client A.

At time t_m , suppose that the client B accesses the data replica stored in its local storage but not been updated by client A's data modification. In order to check the replica consistency, client B first requests the status information of the data replica to the data owner. The data owner finds out that the data set has been modified by client A and requests the data to client A. Client A sends the modified data to the data owner and to the client B, and then waits for the acknowledgement from both. After the data owner updates the modified data set and sets the status field to "done", it sends back an acknowledgement to the client A.

3 Performance Evaluation

In order to measure the performance, we used two Linux clusters, located at Sejong university. Each cluster consists of eight nodes having Pentium3 866MHz CPU, 256 MB of RAM, and 100Mbps of Fast Ethernet each. The operating system installed on those machines was RedHat 9.0 with Linux kernel 2.4.20-8.

The performance results were obtained using the template implemented based on the three-dimensional astrophysics application, developed at the University of Chicago. The total data size generated was about 520MB and among them, 400MB of data were generated for data analysis and data restart, and then the remaining 120MB of data were generated for data visualization. The data sets produced for data visualization are used by the remote clients, thus requiring to perform data replications to minimize data access time.

In Figure 4, we made two client groups, while each group consisting of two nodes. At each time step, a client accesses either 30MB of replicas stored in the local storage, or 30MB of remote data sets stored on the data owner in such a case that the data sets required are not replicated to the local storage.

In the owner-initiated replication, as soon as an application produces data sets at time step 0, all the remote clients receive the necessary visualization data sets to replicate them to the local storage. These replicas are used until the modification to the replicas happens at time steps 5, 10, and 15, respectively.

If the modification to the replicas happens, like occurred at time steps 5, 10, and 15, respectively, the modified replicas are then broadcast to the data owner and to the clients in the sample group, thereby increasing the execution time for accessing remote data sets.

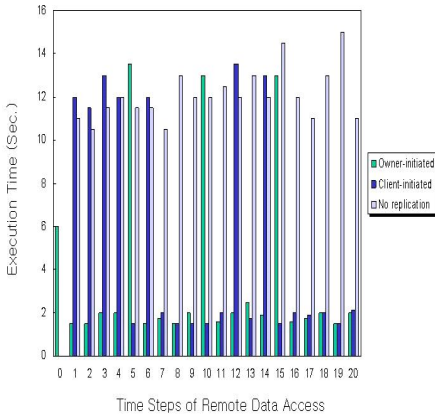


Fig. 4. Execution time for replicating visualization data on the remote clients as a function of time steps for accessing remote data sets. Two client groups were made, while each group consisting of two nodes.

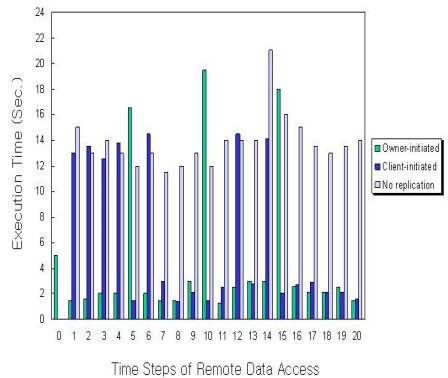


Fig. 5. Execution time for replicating visualization data on the remote clients as a function of time steps for accessing remote data sets. Two client groups were made, while each group consisting of eight nodes.

In the client-initiated replication, since there is no replica stored in the client side until time step 4, each remote client should communicate with the data owner to receive the data sets needed. From time step 5, because each client can use the data replicas stored in its local storage, the execution time for accessing data sets dramatically drops to almost 3 seconds.

When the replicas are modified at time steps 5, 10, and 15, the client-initiated approach just sends to the data owner the IP address of the client modifying the replicas, and thus it takes no more than 3 seconds spent to access metadata. However, we can see that in Figure 4, at time steps 6, 12, and 14, another client tries to access the modified replicas, thus incurring the data communication and I/O costs to update the replicas to the requesting client and to the data owner.

In Figure 5, we increased the number of nodes in each group to eight. Using the owner-initiated data replication, we can see that as the number of nodes in each group is increased the execution time for replicating remote data sets also goes up due to the increment in the communication overhead to broadcast the replicas to the data owner and to the other clients.

On the other hand, as can be seen in Figure 5, the client-initiated data replication shows not much difference in the execution time to receive the modified data sets because less number of nodes than in the owner-initiated data replication is involved in the communication.

4 Conclusion

We have developed two data replication approaches to maintain the replica consistency in case of replica modifications or updates. In the owner-initiated data replication, the replication occurs when applications generate the data sets in the data owner location. In the client-initiated data replication, only when the data sets are needed by a remote client are the necessary data sets replicated to the requesting client. In the future, we plan to use our replication techniques with more applications and evaluate both the usability and performance.

References

1. B. Allcock, I. Foster, V. Nefedova, A. Chervenak, E. Deelman, C. Kesselman, J. Leigh, A. Sim, A. Shoshani, B. Drach, and D. Williams. High-Performance Remote Access to Climate Simulation Data: A Challenge Problem for Data Grid Technologies. SC2001, November 2001
2. M. Cai, A. Chervenak, M. Frank. A Peer-to-Peer Replica Location Service Based on A Distributed Hash Table. Proceedings of the SC2004 Conference (SC2004), November 2004.
3. R. Thakur and W. Gropp. Improving the Performance of Collective Operations in MPICH. In Proceedings of the 10th European PVM/MPI Users' Group Conference (Euro PVM/MPI 2003), September 2003

OntoBayes Approach to Corporate Knowledge

Yi Yang and Jacques Calmet

Institute for Algorithms and Cognitive Systems (IAKS)
University of Karlsruhe (TH)
76131 Karlsruhe, Germany
{yiyang, calmet}@ira.uka.de

Abstract. In this paper, we investigate the integration of virtual knowledge communities (VKC) into an ontology-driven uncertainty model (OntoBayes). The selected overall framework for OntoBayes is the multi-agent paradigm. Agents modeled with OntoBayes have two parts: knowledge and decision making parts. The former is the ontology knowledge while the latter is based upon Bayesian Networks (BN). OntoBayes is thus designed in agreement with the Agent Oriented Abstraction (AOA) paradigm. Agents modeled with OntoBayes possess a common community layer that enables to define, describe and implement corporate knowledge. This layer consists of virtual knowledge communities.

1 Introduction

Corporate knowledge was defined as the overall knowledge detained by agents within a system and their ability to cooperate with each other in order to meet their goal. Decision making based on corporate knowledge has become crucial for a society made of distributed agents each possessing its own knowledge, particularly when the knowledge is uncertain. Uncertainty is an inevitable feature in most environments. Agents do not act within a static and close environment but within a dynamic and open one. The available information is mostly incomplete and often imprecise because agents almost never have access to the whole truth implied by their environment. Agents must therefore act under uncertainty. During a process of decision making agents can profit from the corporate knowledge of their society better than only from their sole knowledge. Agents are required to know not only “what it knows” but also “what they know”, and are expected to make maximum use of the knowledge.

The initial step to build uncertain knowledge-based decision support systems is to adopt a representation. It was solved in an ontology-driven uncertainty model (OntoBayes) [1]. This model proposed an approach to integrate Bayesian Networks (BN) into ontologies, in order to preserve the advantages of both. Bayesian Networks provide an intuitive and coherent probabilistic representation of uncertain knowledge, whereas ontologies can represent the organizational structure of large complex domains excellently.

The next step consists of reasoning under probabilistic knowledge. The Bayesian probability theory is an acknowledged standard for probabilistic reasoning.

It provides not only exact but also approximate reasoning mechanisms. Exact methods exploit the independence structure contained in the network to efficiently propagate uncertainty [2,3]. The approximate methods are based on stochastic simulation, which constitutes an interesting alternative in highly connected networks, where exact algorithms may become inefficient [3,4,5]. OntoBayes should be easily combined with existing reasoning methods to meet its objectives.

The last important step is to construct decision mechanisms. The decision making part of OntoBayes makes use of the classic decision-theoretic techniques. Such techniques possess explicit management of uncertainty and tradeoffs, probability theory and the maximization of expected utility. This part is described in [6].

In this paper, we investigate the integration of virtual knowledge communities (VKC) into OntoBayes, in order to introduce uncertain corporate knowledge in a society of agents. The selected overall framework for OntoBayes is the multiagent paradigm. Agents modeled with OntoBayes have two parts: knowledge and decision making parts. The former is the ontology knowledge while the latter is based upon BN. OntoBayes is thus designed in agreement with the Agent Oriented Abstraction (AOA) paradigm [7]. Agents modeled with OntoBayes possess a common community layer that enables to define, describe and implement corporate knowledge. This layer represents virtual knowledge communities.

The remaining sections of this paper are structured as follows. Section 2 discusses the previous works in this field in further detail. Section 3 provides an overview on the existing approach VKC. Section 4 summarizes the current work on OntoBayes. Section 5 is devoted to investigate the integration of VKC into OntoBayes and finally, Section 6 contains a brief overview on future work and concludes the paper.

2 Previous Works

The work of modeling corporate knowledge based on AOA was a major source of inspiration of this research. The grounding theory behind the design of OntoBayes is the agent oriented abstraction paradigm presented in Calmet's work [7]. The AOA paradigm covers the concepts of agents, annotated knowledge, utility functions and society of agents. Indeed, AOA is based on Weber's classical theory in Sociology [8]. AOA assumes that agents are entities consisting of both a knowledge component and a decision making mechanism. The former is partitioned into four components, also called annotations: ontology, communication, cognition and security. The latter is related to its tasks and goals. It generates utility functions and is based upon the knowledge component. Chiefly, agents can be defined in terms of knowledge and action's utility. The AOA model can be abstractly summarized by a number of basic definitions. A detailed description is to be found in [7].

In [9] the application of the AOA model to the abstract modeling of corporate knowledge is investigated. Corporate knowledge was defined as the amount

of knowledge provided by individual agents. To avoid the separation between agents and knowledge, it was considered that agents have explicitly represented knowledge and communication ability. A knowledge company was modeled as a scenario to demonstrate the corporate knowledge modeling within the AOA.

The concrete implementation for corporate knowledge within AOA was introduced in [10]. The realized concept was introduced as virtual knowledge communities [11,12]. VKC is the applied method to enhance the OntoBayes model for corporate knowledge retrieval, so it is reviewed in the next section separately and in more details.

3 Virtual Knowledge Communities

Traditionally, information is mostly centralized within a uniform information structure. This viewpoint is not truly compliant with the nature of knowledge that is subjective, distributed and contextual [13]. From the perspective of the knowledge information society, modern knowledge management often focuses on the constitution of communities of practice and communities of interest [14].

The concept of a community of practice or a community of interest can be supported in a virtual community in order to bring the concerned agents together to share their knowledge with each other. A community is a place where agents can meet and share knowledge with other agents which share a similar domain of interest. The concept of VKC was introduced as a means for agents to share knowledge about a topic [10]. It aims to increase the efficiency with which information is made available throughout the society of agents.

From the point of view of corporate knowledge management, agents can be individuals, software assistants or automata. Agents possess knowledge and processes within the society tend to make agents produce and exchange knowledge with each other. These processes are distributed throughout the society and contribute through their own intrinsic goals to solve a unique highlevel challenge. This provides the link between corporate knowledge and virtual knowledge communities [9].

In the implemented model [11] there are two main modeling for VKC: agent modeling and community modeling. The former has four key notions: personal ontology, knowledge instances, knowledge cluster and mapper. Personal ontology represents the knowledge of an agent. It describes the taxonomy of the relationships between the concepts and predicates what an agent understands. The knowledge instances are instances of objects defined into the personal ontology. It was assumed that an agent's knowledge consists of both its personal ontology and knowledge instances according to its personal ontology. The knowledge clusters as sub-part of an ontology can be shared among agents. They are defined by their head concept, a pointer to the different parts of knowledge existing in a cluster. The mapper chiefly contains a set of mapping from personal terms to mapped terms, and allows an agent to add such mappings, and use the mapper to normalize or personalize a given knowledge cluster or instance using these mappings. It facilitates knowledge sharing among agents with regards to the heterogeneity of knowledge.

The community modeling has also some key notions: domain of interest, community pack, community buffer. A domain of interest exists in each virtual knowledge community and is similar to the concept of ontology for an agent. It is given by the community leader who created the community. The community pack is what defines the community. It consists of a community knowledge cluster, a normalized ontology which contains at least the head of the community cluster, and the identification of the leaders of the community. The community buffer can record messages which are used by the member of a community to share their knowledge. This approach is compatible with blackboard systems, but still has its difference, because agents cooperate to solve their respective problems, not towards a unique goal.

The VKC approach has been designed and partially implemented as a prototype system. The implementation is based on Java Agent Development Framework (JADE) and Java Runtime Environment (JRE) platform. It was tested and evaluated [11]. A component of the system enables to simulate VKC.

4 Knowledge Representations in OntoBayes

The typical characteristic of domains such as insurance or natural disaster management is the risk factor. The normal ontological model is not sufficient to express the domain specific uncertainty or risks. In order to overcome this shortcoming the OntoBayes model was proposed [1]. This model consists of knowledge and decision making parts. Its decision making part will not be discussed in this paper, because it is not fully relevant to define corporate knowledge within this model. We remind in subsections 4.1 and 4.2 well-known concepts that are basic for our approach.

4.1 Web Ontology Language

OntoBayes makes use of OWL as its formal language to facilitate its knowledge representation. OWL is a semantic markup language for publishing and sharing ontologies on the World Wide Web. It is intended to provide a language that can be used to

- conceptualize domains by defining classes and properties of those classes,
- construct individuals and assert properties about them, and
- reason about these classes and individuals [15].

OWL is developed as a vocabulary extension of the Resource Description Framework (RDF) [16]. The underlying structure of any expression in RDF is a collection of triples, each consisting of a subject, a predicate (also called a property) and an object. OWL and RDF have much in common, but OWL is a language with better descriptive power than RDF.

4.2 Basic Concepts in Probability

The basis element of Bayesian probability is the random variable which is considered as referring to a state (or an event) of the initially unknown environment of

agents. Depending on the type of domain values, random variables are typically divided into three kinds: boolean, discrete and continuous random variables [3]. In the OntoBayes model only discrete random variables are used.

The basic probabilistic concepts used in OntoBayes are the prior and conditional probability. In order to describe what is known about a variable A in the absence of any other evidence Bayesian probability uses the prior (or unconditional) probability. It is often called simply the prior and written as $P(A)$. Once agents have observed some evidence which has influence over the previously random variables, prior probabilities are no longer appropriate. Instead, we use conditional (or posterior) probabilities $P(A|B)$. It indicates the dependency or causal relation between variables A and B : A depends on B (or B causes A).

4.3 Probability-Annotated OWL

To express uncertain information or the risk of actions we need a probabilistic extension of OWL which enables to specify probability-annotated classes or properties. For this purposes we define three OWL classes [1]: “PriorProb”, “CondProb” and “FullProbDist”. The first two classes are defined to identify the prior probability and conditional probability respectively. They have a same datatype property “ProbValue”, which can express the probabilistic value between 0 and 1. The last one is used to specify the full disjoint probability distribution. It has two disjoint object properties: “hasPrior” or “hasCond”.

The property “hasPrior” specifies the relation between classes “FullProbDist” and “PriorProb”. It indicates that the instances of “PriorProb” are elements of one instance of “FullProbDist”. The property “hasCond” describes the relation between classes “FullProbDist” and “CondProb” and it has a similar semantic meaning as “hasPrior”. These two properties are disjoint, because any instance of “FullProbDist” can only have one probability type: either prior or conditional.

4.4 Dependency-Annotated OWL

The probabilistic extension of OWL alone is not enough for modeling our ontology-driven BN. It is required to specify the dependency relations between the random variables explicitly. In order to solve this problem an additional property element `<rdfs:dependsOn>` was introduced in order to markup dependency information in an OWL ontology. Before the relation of dependency is formally defined, it is necessary to introduce some notations which are influenced by the Object Oriented Programming (OOP) approach. We denote an object property s between domain class A and range class B as $s(A, B)$. It is considered as an available operation of the subject class A to object class B . A datatype property d of class A will be denoted as $A.d$, where d is considered as an attribute of A . Now the dependency between properties can be defined as follows.

Definition 1. A dependency is a pair $X \leftarrow Y$, where each of X and Y is either a datatype property $A.d$ or an object property $s(A, B)$. It is read as “ Y depends on X ”.

This definition clearly points out that each random variable in OntoBayes modeled BN is either an object property or a datatype property. The dependency relation was not specified between classes, but between properties, because this avoids possible errors when extracting a Bayesian structure from ontologies [1]. Based upon this definition we can now explain the term of a dependency chain as follows:

Definition 2. A dependency chain is a list $X_1 \leftarrow X_2 \leftarrow \dots \leftarrow X_i \leftarrow \dots$ such that, for each i , $X_i \leftarrow X_{i+1}$ holds. Then X_1 is called root of this dependency chain, and for each i , X_i is an element in this chain.

Now we can build, for each variable X in a dependency chain, a dependency graph for X from a given BN. The graph contains all dependency chains with the condition that every chain contains the element X . There are no dependency circles in any dependency graph, because it is preconditioned that each BN is a directed acyclic graph (DAG).

5 VKC Within OntoBayes

Nowadays most decision support systems put much greater emphasis on the knowledge management. This is clearly justified. Making a decision once one has the right knowledge is often the easy part. Under an open, dynamic and uncertain environment, an agent can make decision more easily, precisely and rationally based upon corporate knowledge than based on the sole knowledge of an agent. OntoBayes facilitates itself as an integral part of a decision support system [1], but does not address social ability of the relevant agents. They can not cooperate and communicate with each other. In order to overcome this limitation the approach of VKC is investigated for enabling corporate knowledge within OntoBayes. The work in [10,11,12] showed the link between corporate knowledge and virtual knowledge communities. Under this proposal we integrate VKC into OntoBayes. This integration aims at equipping agents with a layer through which they have the ability to act as members of knowledge communities. Basically we can use VKC to integrate corporate knowledge in OntoBayes because both of them are ontology-driven models. However there are still some differences due to the fact that the knowledge base of OntoBayes contains Bayesian information besides the ontological one. We illustrate these differences in the sequel and adapt them to the OntoBayes knowledge.

5.1 Differences in Agent Modeling

As mentioned in section 3 by the agent modeling of VKC there are four notions: personal ontology, knowledge instance, knowledge cluster and mapper. In OntoBayes model, the personal ontology is not only the metalevel to the ontology concept level, but also to the Bayesian concept level. Due to this difference we rename this notion as personal OntoBayes. Correspondingly the knowledge instances are the instances of objects defined in the personal OntoBayes, and the concrete probability distribution on the Bayesian level. The knowledge cluster

contains a sub-part of an ontology and a dependency graph with probabilistic information, which can be shared among agents. The mapper is then not only an ontological mapper, but also a Bayesian mapper.

5.2 Differences in Community Modeling

It was pointed out in section 3 that the community modeling has three key notions: a domain of interest, community pack and community buffer. Each VKC has a domain of interest, which is represented with the adapted knowledge cluster as mentioned above. The community pack consists of knowledge cluster, normalized ontology and identification of leaders of this community. Here, except the knowledge cluster, the normalized ontology must also be adapted. Similar to personal ontology, we rename normalized ontology as normalized OntoBayes. It contains at least the head of the knowledge cluster of the community and additionally the dependency graph for all predicates about the head of the knowledge cluster in this community. The community buffer works like a blackboard system, so that it can be smoothly adopted here. It consists of messages following the syntax of the FIPA-ACL standard.

5.3 Knowledge Exchange with an Example

The main objective to use VKC in OntoBayes is to enable the knowledge exchange between individual agents. Table 1 shows what are the exchanges in VKC, particularly in the OntoBayes model. While only the ontological knowledge can be exchanged in normal VKC, Bayesian knowledge must be also exchanged additionally.

Table 1. The exchanged knowledge in OntoBayes model

	Knowledge cluster	Knowledge instance
Bayesian knowledge	Bayesian Networks	Instances of BN and their Probability distribution
Ontological knowledge	Ontological structure	Instances of ontology

We give an example in Fig. 1 to show how knowledge exchange with VKC works in OntoBayes. The dashed-arrowed line shows the operation in an exchange process and the solid arrowed line shows the dependency relation between properties. The notation mentioned in section 4.4 is used in this example. There is a community with a domain of interest “insurance product” in the community of communities. An agent “insurance” joined this community and wrote a message in the community buffer. This message indicates that the premium of an insurance product depends on the term and risk coverage of the product. Besides this agent there is another agent “customer” who has also interest in this community and writes a message. This message indicates that an action for buying an insurance product depends on its premium. After the message

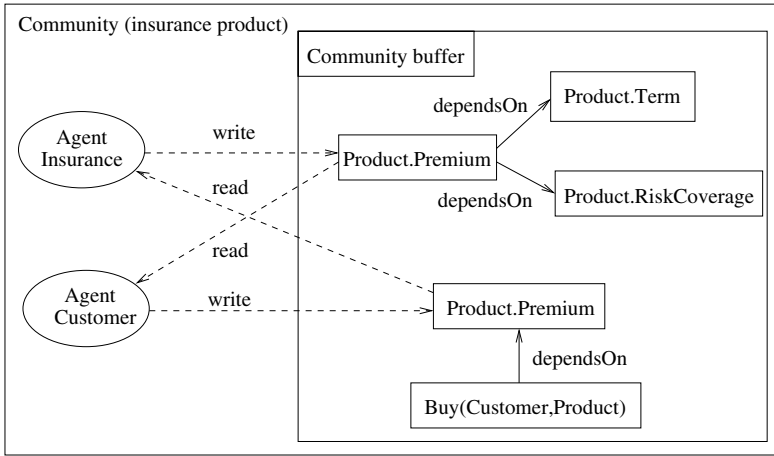


Fig. 1. A simple knowledge exchange example

input both of them read the messages from other agents in this community, to complete a simple knowledge exchange process.

After the exchange each agent can adapt its old knowledge base respectively to its knowledge cluster and personal OntoBayes with the new information. The feature introduced in the above example is illustrated in Fig. 2.

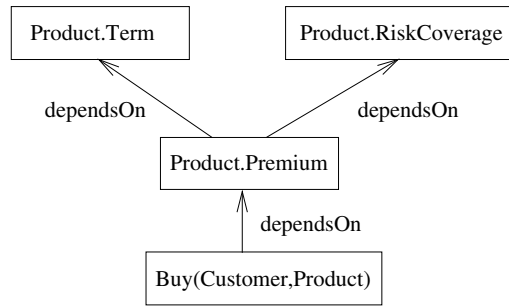


Fig. 2. Knowledge evolvement following the knowledge exchange

It must be pointed out that not only the structure of Bayesian Networks , but also the probability distribution according to the structure will evolve after the knowledge exchange. To assess the evolution of the probability distribution, methods such as conditional independency, Markov blanket etc. were introduced in [17]. The guarantee of knowledge sharing will not be investigated in this paper.

5.4 Knowledge Community Processes

Agents' actions related to knowledge communities are the following ones: initiate, reorient, leave, terminate and join a community as well as exchange knowledge.

Every agent can initiate a community by creating a topic and a community buffer and advertising about this community. Advertising is done through a specific agent called “community of communities”, which has a central directory of all communities. All agents of the system are members of this community. At the same time of initiating a community, a community park is also created. It contains a knowledge cluster of the initiator, a normalized OntoBayes specified above, and the identification of the initiators. The information will be posted to the community buffer.

Community reorientation is needed because the knowledge can not be uniquely considered at design time. It should evolve over time. Reorientation consists of sending a new community cluster to the community of communities.

Agents can leave a community voluntarily, but it could be also forced out by the leaders. When a leader leaves a community, a new leader is required, if it is the unique leader in this community.

Community termination consists of erasing the community buffer and its reference posted to the community of communities during the community’s lifetime. The community can only be terminated by one of the community leaders.

6 Conclusion and Future Work

This paper aims to give an overview of an ongoing, but long term, research project. There is apparently no similar work in the research domain of decision support system under uncertain knowledge. The paper builds upon the first version of the OntoBayes model [1] and the work of virtual knowledge communities [10,11,12], to model corporate knowledge under an open, dynamic and uncertain environment. OntoBayes provides an efficient way for representing uncertain knowledge in an ontological and Bayesian approach. The grounding theory for OntoBayes and VKC is the Agent Oriented Abstraction, which can describe in a fully generic way the concept of an agent and the society of agents. It is showed that such an abstraction mechanism leads to very practical applications for corporate knowledge [9,10]. It was investigated in this paper that virtual knowledge communities can be applied to OntoBayes with some adaptations, because of the similarity: both of them are ontology-driven model, even though they differ with each other on the agent’s personal knowledge, knowledge instance and knowledge cluster.

Although the VKC enhanced OntoBayes model in this paper is a considerable step towards corporate knowledge based decision support system, further investigations are still required. A system is being implemented for applications in insurance and natural disaster management. Some of the future research directions deal with:

- A trust mechanism for agents involved in uncertain knowledge sharing.
- Methods to assess achieved decisions based upon corporate knowledge with possible conflicting information.

References

1. Yang, Y., Calmet, J.: Ontobayes: An ontology-driven uncertainty model. In: *Proceeding of the International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC'05)*. Volume I, Vienna, Austria, IEEE Computer Society (2005) 457–464
2. Pearl, J.: *Probabilistic reasoning in intelligent systems*. 2 edn. Morgan Kaufmann (1997)
3. Russell, S.J., Norvig, P.: *Artificial intelligence: A Modern Approach*. 2. ed. edn. Prentice Hall (2003)
4. Shachter, R., Peot, M.: Simulation approaches to general probabilistic inference on belief networks. In: *Proceedings of the 5th Annual Conference on Uncertainty in Artificial Intelligence (UAI'90)*, New York, NY, Elsevier Science Publishing Company, Inc. (1990) 221–234
5. Bouckaert, R.R., Castillo, E., Gutiérrez, J.M.: A modified simulation scheme for inference in bayesian networks. *International Journal of Approximate Reasoning* **14** (1996) 55–80
6. Yang, Y., Calmet, J.: Decision making in ontology-based uncertainty model. 21st *European Conference on Operational Research* (2006)
7. Calmet, J., Maret, P., Endsuleit, R.: Agent-oriented abstraction. *Revista (Real Academia de Ciencias, Serie A de Matematicas)* **98** (2004) 77–84 Special Issue on Symbolic Computation and Artificial Intelligence.
8. Weber, M.: *Economy and society*. University of California Press (1986)
9. Maret, P., Calmet, J.: Modeling corporate knowledge within the agent oriented abstraction. In: *Proc. International Conference on Cyberworlds (CW'04)*, IEEE Computer Society (2004) 224–231
10. Maret, P., Hammond, M., Calmet, J.: Virtual knowledge communities for corporate knowledge issues. In: *Proceedings 5th International Workshop on Engineering Societies in the Agents World (ESAW'04)*. Volume 3451 of *Lecture Notes in Computer Science*, Toulouse, France, Springer (2004) 33–44
11. Hammond, M.: *Virtual knowledge communities for distributed knowledge management: A multi-agent-based approach using jade*. Master's thesis, University of Karlsruhe and Imperial College London (2004)
12. Maret, P., Calmet, J.: Corporate knowledge in cyberworlds. *IEICE Transaction, Information and Systems Society* **E88-D** (2005) 880–887
13. Bonifacio, M., Bouquet, P., Cuel, R.: Knowledge nodes: the building blocks of a distributed approach to knowledge management. *Journal of Universal Computer Science* **8** (2002) 652–661
14. Fischer, G., Ostwald, J.: Knowledge management: Problems, promises, realities, and challenges. *IEEE Intelligent Systems* **16** (2001) 60–72
15. W3C: (OWL Web Ontology Language Guide) <http://www.w3.org/TR/owl-guide>.
16. W3C: (Resource Description Framework (RDF): Concepts and Abstract Syntax) <http://www.w3.org/TR/rdf-concepts>.
17. Jensen, F.V.: *An introduction to Bayesian networks*. UCL Press (1996)

About Inclusion-Based Generalized Yes/No Queries in a Possibilistic Database Context

Patrick Bosc, Nadia Lietard, and Olivier Pivert

IRISA/Enssat, Technopole Anticipa BP 80518,
22300 Lannion, France
{bosc, lietard, pivert}@enssat.fr

Abstract. This paper deals with the querying of possibilistic relational databases, by means of queries called generalized yes/no queries, whose general form is: "to what extent is it possible that the answer to Q satisfies property P". Here, property P concerns the inclusion in the result, of a set of tuples specified by the user. A processing technique for such queries is proposed, which avoids computing the worlds attached to the possibilistic database.

1 Introduction

In this paper, we consider relational databases where some attribute values are imprecisely known. Different formalisms can be used to represent imprecise information (see for instance [4, 6]), and the possibilistic setting is assumed in the rest of the paper. Possibility theory [7] provides an ordinal model for uncertainty where imprecision is represented by means of a preference relation encoded by a total order over the possible situations. This approach provides a unified framework for representing precise values, as well as imprecise ones (regular sets) or vague ones (fuzzy sets), and various null value situations (see [4] for more details).

A key question is to define a sound semantics for queries addressed to imprecise databases. Since imprecise data are represented as sets of acceptable candidates, an imprecise database can be seen as a set of regular databases, called worlds, associated with a choice for each attribute value. A naïve query processing method would consist in computing the result of the query in each world of the database concerned, but this is clearly intractable due to the huge number of worlds. This observation leads to consider only specific queries which can be processed directly against the possibilistic database, while delivering a sound result. A compact calculus valid for a subset of the relational algebra has thus been devised [2] and, in this context, the result of a query is a possibilistic relation whose interpretations correspond to more or less possible results, equivalent to those which would be obtained with a world-based calculus.

This achievement is interesting from a methodological point of view, but the use of this type of result by an end-user can be somewhat delicate. So, it is convenient to define queries which are more specialized to fit user needs. In this paper, we consider generalized yes/no queries (concept inspired by that introduced by Abiteboul [1] in a context of null values), whose form is: "to what extent is it possible and certain that the answer to Q satisfies property P". We already studied in [5] the case where P is:

"contains a given (specified) tuple t ", and here, we deal with inclusion-based possibilistic queries where property P is: "contains the tuples $t_1, t_2, \dots$ and t_k ". To the best of our knowledge, there does not exist any previous work about this issue.

The structure of the paper is the following. In section 2, the notion of a possibilistic relational database is introduced. Then, the data model requested for a valid compact processing of algebraic queries is briefly presented. In section 3, we propose a "trial and error" algorithm aimed at processing inclusion-based possibilistic queries and we briefly discuss the complexity of this approach. Finally, the conclusion summarizes the contributions of the paper and draws some lines for future works.

2 A Possibilistic Database Model

From a semantic point of view, a possibilistic database D can be interpreted as a set of usual databases (worlds), denoted by $\text{rep}(D)$, each of which being more or less possible (one of them is supposed to correspond to the actual state of the universe modeled). Any world W_i is obtained by choosing a candidate value in each possibility distribution appearing in D and its possibility degree is the minimum of those of the candidates chosen (according to the axioms of possibility theory since choices are assumed to be independent). In the following, only finite distributions are considered.

A calculus based on the processing of the query Q against worlds is intractable and a compact approach to the processing of Q must be found out. It is then necessary to be provided with both a data model and operations which have good properties: i) the data model must be closed for the considered operations, and ii) it must be possible to process any query in a compact way. In addition, its result must be a compact representation of the results obtained if the query were applied to all the worlds drawn from the possibilistic database D , i.e.: $\text{rep}(Qc(D)) = Q(\text{rep}(D))$, where Qc stands for the query obtained by replacing the operators of Q by their compact versions. This property characterizes data models called strong representation systems. A data model complying with this property was proposed in [2], which involves usual possibilistic relations enriched with: 1) a degree N associated with every tuple, which expresses the certainty that the tuple has a representative in any world and 2) nested relations used to model possibility distributions over several attributes).

The algebraic operators which can be processed in a compact way within this model are the selection, projection, union and a specific join called fk -join (whose definitions can be found in [2]). Hereafter is an example of a compact possibilistic relation (involving the nested attribute $X(\text{date}, \text{place})$), considered to be the result of the compact processing of a given algebraic query:

res	#i	ap	X	N
			date	place
	i_1	B-727	$\{1/\langle d_1, c_1 \rangle + 0.7/\langle d_1, c_2 \rangle + 0.4/\langle d_3, c_2 \rangle\}$	1
	i_3	B-727	$\langle d_1, c_2 \rangle$	0.3
	i_4	B-727	$\{0.3/\langle d_3, c_2 \rangle\}$	0

This relation is associated with 12 worlds since the first tuple admits 3 interpretations, the second and third ones have 2 interpretations among which $\emptyset$ (no representative).

3 Inclusion-Based Generalized Yes/No Queries

3.1 Principle

We are interested in inclusion-based yes/no queries whose general form is: "to what extent is it possible and certain that tuples $t_1, t_2, \dots$ and t_k belong to the result of the algebraic query Q ?". Such queries are an extension of those studied in [3, 5].

The evaluation we propose for such queries is based on a four-step mechanism:

- 1) a pre-processing aiming at reducing the volume of intermediary results. It consists in discarding the attributes which do not appear in Q .
- 2) a compact processing of the associated algebraic query, which builds a compact relation res according to the procedure outlined in the preceding section,
- 3) a post-processing consisting in performing a selection which removes from res the associations of candidate values that do not correspond to any target tuple. This will reduce the complexity of step 4. If the schema of relation res is $RES(A_1, \dots, A_p)$, the corresponding selection is:

$$((A_1 = t_1.A_1 \text{ and } \dots \text{ and } A_p = t_1.A_p) \text{ or } \dots \text{ or } (A_1 = t_k.A_1 \text{ and } \dots \text{ and } A_p = t_k.A_p))$$

This selection leads to a nested relation res' such that each of its tuples can produce at least one of the target tuples $t_1, t_2, \dots$ and t_k .

- 4) computation of the final degrees Π and N . The first one is computed thanks to a "trial and error" procedure whose principle is to explore the worlds of res' but in a non-exhaustive way (thanks to some pruning conditions) in order to construct the best vector V , i.e., the most possible interpretation of res' containing all the target tuples. It consists in generating for each tuple u_i of relation res' the set E_i of its representatives x_j (precise tuples), to which the empty set $\emptyset$ is added if $N < 1$, and the set Π_i of their respective possibility degrees π_j ($1 - N$ if x_j is $\emptyset$). These sets are then ranked in decreasing order on the possibility degrees attached to the tuples x_j (see remarks below). Then, a tuple is chosen in each set E_i at each step of the procedure in order to form vector V .

As to the necessity degree, it is computed the following way:

- if the possibility degree computed thanks to the "trial and error" algorithm is less than 1, the necessity degree will be 0 (due to the property $\Pi(A) < 1 \Rightarrow N(A) = 0$);
- else, one has to compute $N(\{t_1, \dots, t_k\} \subseteq res(Q))$. According to the axioms of possibility theory, this degree is equal to

$$\begin{aligned} N(t_1 \in res(Q) \text{ and } \dots \text{ and } t_k \in res(Q)) &= 1 - \Pi(t_1 \notin res(Q) \text{ or } \dots \text{ or } t_k \notin res(Q)) \\ &= 1 - \max(\Pi(t_1 \notin res(Q)), \dots, \Pi(t_k \notin res(Q))) \\ &= 1 - \max(1 - N(t_1 \in res(Q)), \dots, 1 - N(t_k \in res(Q))). \end{aligned}$$

In [5], we proposed a linear complexity algorithm in order to compute the necessity that a given tuple t belongs to the result of Q ($N(t \in res(Q))$). Thus, the computation of N in the case considered here amounts to evaluating k times this algorithm.

The procedure for the computation of the possibility degree is presented hereafter:

```

procedure OptimalSolution (i integer)
begin
  compute  $E_i$ ;
  for  $x_i$  in  $E_i$  do
    if satisfactory( $x_i$ ) then
      memorize( $x_i$ );
      if solutionFound then
        if better then keepSolution endif
      else if stillPossible then
        optimalSolution(i + 1) endif
      endif;
      undo( $x_i$ );
    endif;
  endfor;
end;

```

The different components of this algorithm are:

Pos: vector of dimension n , it contains the possibility degrees of the tuples of V ;

B: vector of booleans, $B[q]$ is true iff the target tuple t_q has been found and belongs to the current world under construction;

Card: the number of target tuples found so far in the world under construction;

Best Π : the possibility degree of the most possible world found so far;

satisfactory(x_j): $\pi_j > \text{Best}\Pi$;

memorize(x_j): $V[i] \leftarrow x_j$; $\text{Pos}[i] \leftarrow \pi_j$;

for $q = 1$ to k do

if $x_j = t_q$ then

$B[q] \leftarrow \text{true}$; if (x_j is not already in V) then $\text{Card} \leftarrow \text{Card} + 1$ endif;

endif;

enddo;

solutionFound:

(($i = n$) and ($B[1]$ and ... and $B[k]$)); the tuples of relation res' have been all examined, and all the target tuples belong to the current world.

better: $\min_{q \text{ in } [1, n]} \text{Pos}[q] > \text{Best}\Pi$;

keepSolution: $\text{Best}\Pi \leftarrow \min_{q \text{ in } [1, n]} \text{Pos}[q]$;

stillPossible: $\text{Card} + (n - i) \geq k$ (k being the number of the target tuples);

undo(x_j):

if it exists q such that $x_j = t_q$ and (x_j is not already in V)

then $\text{Card} \leftarrow \text{Card} - 1$; $B[q] \leftarrow \text{false}$ endif;

endif;

Remarks. In order to reduce the number of worlds to be computed, some improvements can be brought to the above algorithm:

- 1) When $Best\Pi$ is equal to 1 the processing can be stopped.
- 2) The sets E_i can be ranked in decreasing order on the possibility degrees. In that case, once an unsatisfactory x_j is found, the loop can be stopped (because the following x -values would be unsatisfactory too).
- 3) We can take advantage of the number of tuples of relation res' . Indeed, if relation res contains a number of tuples which is less than k (the number of the target tuples), the result would obviously be $\Pi = N = 0$.

3.2 Example and Performances

Let us consider the following relation res which is supposed to be the result of the evaluation of a given query Q :

res	lg	date
	20	$\{1/d_3 + 0.7/d_1\}$
	$\{1/20 + 0.7/25\}$	$\{1/d_1 + 0.6/d_2\}$
	$\{0.3/25 + 1/18\}$	d_2
	$\{1/20 + 0.5/28\}$	$\{0.2/d_1 + 1/d_4\}$

and the query: "to what extent is it possible and certain that $t_1 = \langle 20, d_1 \rangle$, $t_2 = \langle 25, d_2 \rangle$ and $t_3 = \langle 20, d_3 \rangle$ belong to the result of the query Q ?". The result of the selection (($lg = 20$ and $date = d_1$) or ($lg = 25$ and $date = d_2$) or ($lg = 20$ and $date = d_3$)) is:

res'	X	N
	lg	date
	$\{0.7/\langle 20, d_1 \rangle + 1/\langle 20, d_3 \rangle\}$	1
	$\{1/\langle 20, d_1 \rangle + 0.6/\langle 25, d_2 \rangle\}$	0.3
	$\{0.3/\langle 25, d_2 \rangle\}$	0
	$\{0.2/\langle 20, d_1 \rangle\}$	0

Thanks to this selection, each tuple of relation res' can only generate the target tuples t_1 , t_2 and t_3 , and the number of worlds has decreased (24 instead of 64). The result returned by the trial and error procedure is $\Pi = 0.3$, thus we do not need to call on the second algorithm since we can conclude that $N = 0$.

Concerning the complexity aspect, several remarks can be made. The algorithm computing Π includes two pruning conditions: one based on the optimality of the solution, the other on the cardinality of the current world under construction. Clearly, the first condition will be very effective if a highly possible satisfactory world is encountered early. If it is not the case, only the second pruning condition can have an effect. Let us notice, however, that the number of worlds attached to res is much smaller than the number of worlds attached to the initial possibilistic database, due to the reduction obtained through the compact evaluation of query Q . Anyway, when the cardinality of res is too important, the complexity will still be too high to reach acceptable performances. Hence, it is important to envisage some ways to overcome this problem. A first idea is to compute the necessity degree N first (which is not too expensive since the corresponding algorithm has a linear complexity), and to compute the possibility degree only when N equals zero (since, according to possibility theory,

$N > 0 \Rightarrow \Pi = 1$). A second idea consists in computing only an approximate answer, i.e., an underestimation of the actual value of Π . This could be done by means of a greedy algorithm scanning relation res' and choosing, for each imprecise tuple of res' , the most possible candidate equal to a target tuple not found yet.

4 Conclusion

This paper deals with the querying of possibilistic relational databases, by means of queries called generalized yes/no queries, whose general form is: "to what extent is it possible that the answer to Q satisfies property P ". More precisely, inclusion-based queries, where property P concerns the inclusion in the result of a set of tuples specified by the user, have been investigated. The possibility and necessity degrees, which constitute the answer to these queries, are respectively computed thanks to a "trial and error" procedure and an algorithm of linear complexity.

This work opens different lines for future research. One of them is related to the efficiency of the method. Clearly the approach proposed is much more efficient than a technique based on the computation of worlds but it would be interesting to assess experimentally the performances of the solutions outlined at the end of section 3. Another direction concerns the impact of the model of uncertainty on the feasibility and performances of the technique proposed here. More precisely, one may wonder if the evaluation method described in this paper would still be valid in a probabilistic database framework, and if so, if it would be more or less efficient in such a context.

References

1. Abiteboul, S., Kanellakis, P., Grahne, G.: On the Representation and Querying of Sets of Possible Worlds. *Theoret. Comput. Sci.* 78 (1991) 159–187
2. Bosc, P., Pivert, O.: About Projection-selection-join Queries Addressed to Possibilistic Relational Databases. *IEEE Transactions on Fuzzy Systems*. 31 (2005) 124–139
3. Bosc, P., Lietard, N., Pivert, O.: About Cardinality-based Possibilistic Queries against Possibilistic Databases. *EUSFLAT-LFA*. (2005) 1176–1275
4. Bosc, P., Prade, H.: An Introduction to the Treatment of Flexible Queries and Uncertain or Imprecise Databases. *Uncertainty Management in Information Systems*, A. Motro and P. Smets (Eds), Kluwer Academic Publishers. (1997) 285–324
5. Bosc, P., Pivert, O.: About Yes/No Queries against Possibilistic Databases. *International Journal of Intelligent Systems*, to appear.
6. Imielinski, T., Lipski, W.: Incomplete Information in Relational Databases. *Journal of the ACM*. 31 (1984) 761–791
7. Zadeh, L. A.: Fuzzy Sets as a Basis for a Theory of Possibility. *Fuzzy Sets and Systems*. 1 (1978) 3–28

Flexible Querying of an Intelligent Information System for EU Joint Project Proposals in a Specific Topic

Stefan Trausan-Matu, Ruxandra Bantea, Vlad Posea,
Diana Petrescu, and Alexandru Gartner

“Politehnica” University of Bucharest
Computer Science Department
313, Splaiul Independentei
Bucharest, Romania
trausan@racai.ro
<http://www.racai.ro/~trausan>

Abstract. This paper presents a semantics-driven web portal allowing browsing and flexible querying state-of-the-art data on organizations, national and international projects, current research topics, technologies and software solutions in the domain of the European Research in the FP6/IST program. The backbone of this portal consists of an ontology-based knowledge base structuring information about the above-described area. The portal incorporates advanced information retrieval methods, based on ontologies and on natural language processing.

1 Introduction

The large amount of information that is daily changing and growing on the web is a challenge for communities of practice that need to be informed with the latest news in their domain of interest. In such a case, web search engines do not always perform in the expected way, especially because they are designed as general purpose tools, mainly based on keyword searching in any domains. An unfocused approach often leads to undesired results: a lot of uninteresting documents are returned and very important data could be missed. In order to allow more intelligent information retrieval, that exploits the “unwritten” facts, the community of users can take advantage of domain-specific web Portals that are structured around of an ontology for that domain. We consider this approach as one that can lead to a better focused search.

The present paper describes a knowledge-based portal for a community of researchers that is preparing project proposals in a given domain for European research programs like IST/FP6¹. The information of interest for this community are the potential partners, existing research projects, and new call for project proposals, all of them belonging to specific topics².

The target users of the portal are researchers in the defined area, members of universities and other organizations, which are involved in projects in the given field. For

¹ This portal was developed in the EU-NCIT project, an EU funded Center of Excellence at the Politehnica University of Bucharest.

² In EU-NCIT: semantic web, e-learning, grid systems, and collaborative work.

this community of people, various methods to organize and especially to find the appropriate information are needed. Users with greater experience with the content of the portal can use some kind of search methods while occasional visitors, that, for example, simply explore the possibilities of the IST program, would probably prefer other, more general query methods, using natural language.

The development of the project started with studying other similar domain specific Web Portals and analyzing the underlying technology that was used, in order to extract and compare different design approaches. Some of these portals have implemented only a browsing module, only few of them offering also a search facility. We chose to implement a complex searching module, the solution being further explained in the remainder of the paper. Therefore, at the core of the portal is an ontology that plays the role of the declarative knowledge repository containing the basic concepts. The knowledge-based technology allows flexible querying, suitable to each user's profile. In addition, other issues were also considered: the introducing of a level of security to the portal, managed by user accounts, and of a news service, useful in those portals that addressed to a large community of users, and in a dynamic field, with a lot of events, seminars, info days etc.

The next section discusses some state of the art aspects on intelligent, knowledge-based portals. The third section presents the EU-NCIT Portal that has been developed at the Politehnica University of Bucharest. The section before conclusions describes the flexible query engines available in EU-NCIT.

2 Knowledge-Based Web Portals

There are some problems that should be taken into consideration when creating a web portal. For example, due to their generality and keyword-based algorithms, classical search engines usually return a lot of uninteresting documents while others are missed because they do not contain the provided keywords.

In order to allow a flexible, intelligent information retrieval, that exploits the "un-written" facts in documents, that cannot be discovered by a keyword-based search, web portals should be based on the definition of an ontology of the considered domain. The ontology should describe the terminology corresponding to the content and some knowledge that can help inferring new relations and terms based on others. For example, the ontology might include concepts such as "software projects", "programming languages", "organization" or "people". In this ontology there must be available also knowledge that state that "all software projects are created using programming languages", or that "people work in organizations", "organizations produce software projects". These definitions should allow other true facts to be inferred, so that, the users will obtain better search results from the portal that are practically impossible to obtain from conventional retrieval systems.

There is not a lot of experience in the development of knowledge-base based portals. The approaches differ in the methods used in defining the ontology structure, the population of knowledge base, or querying the knowledge base [4]. For example, the current version of the Ontoweb portal [6] and its underlying core methodology SEAL (SEmantic portAL) [7] support both (conceptual) browsing and querying of the information stored in the portal, as well as content provision and access.

Another project based on KAON portal is the VISION PORTAL³, which describes the European Knowledge Management community. The backbone of this portal, developed in the context of a EU-funded Strategic Roadmap project, consists of an ontology-based knowledge base structuring information and knowledge about the Knowledge Management area.

An overview of semantic web portals is given in [7]. Besides the VISION project, several other EU-funded Projects also focused on developing a KM-related Strategic Roadmap (e.g., Rocket6 - Roadmap to Communicating Knowledge Essential for the industrial environment, VOMAP - Roadmap Design For Collaborative Virtual Organizations in Dynamic Business Ecosystems, COCONET - Context-Aware Collaborative Environments for Next Generation Business Networks).

3 The Structure of the EU-NCIT Portal

The EU-NCIT Portal allows browsing and querying the state-of-the-art of organizations, national and international projects, current research topics, technologies and software solutions in the domain of the European Research in the FP6/IST program. The backbone of this portal consists of an ontology-based knowledge base structuring information about the above-described area, allowing intelligent retrieving of information. The solution presented in this paper uses the KAON infrastructure as the systems mentioned in the previous section also do.

The features that the system provides include access to a collection of files with useful data for the community of users, administrative aspects such as membership management, a news notification service, and an intelligent search engine, based on the knowledge base.

The knowledge base is constructed on the structure of an ontology of European projects. This ontology is based on the model of an European project that implicitly appears in the FP6/IST website [2]. According to that model, seven basic concepts have been identified at the top level of our ontology: Project, Call for proposal, Topic, Instrument, Event, Organization, and Person. These concepts are linked by a large number of relations. For example, the project concept has relations with almost every other concept. These relations are "Project has topic", "Project is addressed in call", "Project main organization", "Project has participants", "Project has instrument", "Has contact person".

Knowledge acquisition (KA) is one of the most important activities in knowledge engineering, because it is the main manner of building a knowledge base. KA is performed in our approach through a crawler implemented in Java, a stand-alone application that runs in the background, from time to time, and whose objective is to update the website by modifying the ontology.

The knowledge-based solution on organizing the portal offers the opportunity to consider similarity measures and making inferences for generating new relations. Such measures can detect how close two concepts are, based usually on semantic distances [5,8,9]. These measures may show, for example, how close two organizations are, based on common projects or based on projects on common areas. Some other

³ The VISION-portal can be assessed on <http://wim.fzi.de:8080/visionportal/dispatcher/>

examples are finding similar projects based on similarity measures instead of common keywords, identifying communities of persons with common interests based also on their participations in similar projects. The similarity measures will be implemented in the future extensions of the application.

A second advantage of using ontologies is the possibility of identification of new relations based on inferences starting from the existing knowledge. The inferences may be performed by a reasoning engine, using programs in the Jess language [3], or classifiers in Descriptions Logics [1] associated with ontology environments.

4 Flexible Search in the EU-NCIT Portal

The search module in EU-NCIT helps users to find information in a flexible way. The search engines that have been implemented were designed based on the structure of the ontology.

Four different types of searching have been implemented in the portal. The first type of search, based on relations, performs very well when people want to find:

- Projects on different topics, developed by different organizations
- Organizations that have worked on some topics, on some projects, or that have experienced with different calls or instruments
- Contact persons for topics or projects
- Persons that have experienced with some topic, organization, project, call or instrument.

The user can choose to query instances of one type of concept, based on the relations that are displayed for him in its dropping menu. In addition to choosing relations modeled in the ontology, the user may also query inferred relations that are not explicitly stored in the knowledge base.

The second type of search is using a module that browses the ontology. The ontology skeleton is seen as a tree and its nodes are hyperlinks referring to other concepts or to instances. Starting from the main concepts, the user can get particular information about any instance of any other concept.

For example, if the user desires to find out who is the person responsible for a known project, he can access the PROJECT concept; can find the instance of that project, and going deeper on the hyperlinks he can find out who is the contact person for that project.

Keywords search, the third type of search is different from the previous two, mainly because of the way the ontology is handled. In the other two options, the ontology was accessed using a query language and an inference module. The relations and the structure of the ontology played an important part in the information retrieval. In the “keyword model”, the user is entering a set of keywords (which may not be present in the ontology) to find matching projects.

The natural language querying module has the goal to help visitors that are at their first visits on the portal to find the appropriate information. These users are not familiar at all with the knowledge base, with the concepts and their relations. For them, the existence of those elements is transparent. The natural language-based query method takes advantage of the ontology system, but the user is obviously not aware of that.

He simply supplies some questions and gets the answer in the standard output format that the Portal uses for answering the queries.



Fig. 1. NL-Query Page

The algorithm starts by reading the full questions from the user. Then, after some preliminary processing is done to this input (e.g. eliminating stop words that have no significance) the module looks for the presence of some of the known concepts from the ontology. After this is done, the relations between concepts must be guessed from this input, in order to get the answer. The algorithm is based on the domain ontology and on the WordNet (<http://wordnet.princeton.edu>) lexical ontology.



Fig. 2. The NLP-module's answer

5 Conclusions

The paper presents a knowledge-based portal that was implemented using semantic web technologies (ontologies, production rules, information retrieval, XML-based mark-up languages, KAON, JENA and RDQL, and natural language processing). The

portal is functional, it is used in the EU-NCIT community and it demonstrated that is a feasible technology. New facilities are permanently included for extending the inference capabilities, using the Jess [3] language for production rules.

References

1. F. Baader, I. Horrocks, and U. Sattler, "Description logics as ontology languages for the semantic web", in *Festschrift in honor of Jorg Siekmann, D. Hutter and W. Stephan*, Eds. LNAI, Springer-Verlag, 2003,
2. FP6 – IST web site – Available: <http://www.cordis.lu/ist/>
3. E. Friedman-Hill, "Jess in Action: Rule-Based Systems in Java" Manning Publications, 2003
4. M. Hefke, "VISION - a Semantic Web Portal for Describing the State-of-the-art on European Knowledge Management". LWA 2004: 245-251
5. G. Hirst, D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms". in Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, chapter 13, pages 305 - 332. The MIT Press, Cambridge, MA, (1998).
6. Spyns, D. Oberle, R. Volz, J.Zheng, M. Jarrar, Y. Sure, R.Studer, R. Meersman. *OntoWeb – a Semantic Web Community portal*, Proceedings of the Fourth International Conference on Practical Aspects of Knowledge Management (PAKM 2002), Vienna, Austria, 2002
7. Staab, R. Studer, Y. Sure, R. Volz. *SEAL - a SEMantic portAL with content management functionality*. In: *Gaining Insight from Research Information. CRIS 2002 - Proceedings of the 6th Conference on Current Re-search Information Systems*, August 29-31, 2002, Kassel, Germany, 2002
8. M. J. Sussna, "Text Retrieval Using Inference in Semantic, Metanetworks" PhD thesis, University of California, San Diego, (1997)
9. Z. Wu, M. Palmer, "Verb semantics and lexical selection", in *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133-138, Las Cruces, New Mexico, June (1994).

Multi-expression Face Recognition Using Neural Networks and Feature Approximation

Adnan Khashman¹ and Akram A. Garad²

^{1,2}Electrical & Electronic Engineering Department, Near East University, Nicosia, Cyprus
amk@neu.edu.tr, akram_garad@yahoo.com

Abstract. A human face is a complex object with features that can vary over time. Face recognition systems have been investigated while developing biometrics technologies. This paper presents a face recognition system that uses eyes, nose and mouth approximations for training a neural network to recognize faces in different expressions such as natural, smiley, sad and surprised. The developed system is implemented using our face database and the ORL face database. A comparison will be drawn between our method and two other face recognition methods; namely PCA and LDA. Experimental results suggest that our method performs well and provides a fast, efficient system for recognizing faces with different expressions.

1 Introduction

Face recognition by machines can be invaluable and has various important applications in real life, such as, electronic and physical access control, homeland security and defense. Many methods have been developed to provide machine recognition of human faces, such as Template Matching [1], Principle Component Analysis (PCA) [2], Linear Discriminant Analysis (LDA) [3] and Independent Component Analysis (ICA) [4]. PCA and LDA are considered as linear subspace learning algorithms and focus on the global structure of the Euclidean space. ICA is a statistical method for transforming an observed multidimensional random vector into its components that are statistically as independent from each other as possible.

Recent works suggested different methods for face recognition, such as using Eigenphases [5], discriminant analysis-based systems [6], [7], Multiblock-fusion scheme [8] and local regions appearance-based recognition [9]. The use of neural networks for face recognition has also been recently addressed [10], [11], [12].

The work that is presented within this paper aims at developing an intelligent face recognition system that recognizes multi-expression faces. The developed system uses a back propagation neural network that learns approximations of essential face features from different facial expressions. The features are the eyes, nose and mouth which are approximated by averaging the feature vectors from four face expressions; natural, smiley, sad and surprised. Averaging creates a "fuzzy" feature vector as compared to multiple "crisp" feature vectors. The averaged feature vectors are presented to the neural network as pattern vectors during the training phase. For testing the trained neural network, no feature averaging is applied to the face images.

Our suggestion is that, the trained neural network identifies the persons regardless of their different face expressions by learning the averaged approximations of the essential face features which may vary from one expression to another. The use of four expressions for training the neural network provides sufficient data for training the neural network to identify the persons.

This paper presents an efficient intelligent face recognition system using face feature approximation and a neural network. Successful results have been achieved, and will be compared to using PCA and LDA face recognition methods. The structure of the paper is as follows: in section 2 a brief explanation of the face image databases is presented. In section 3 the feature approximation method is described showing face image preprocessing, feature averaging and neural network implementation. In section 4 the results of training and testing the neural network using the face image databases are presented, and a comparison to other face recognition methods is drawn. Finally, section 5 concludes the work and suggests future work.

2 Face Databases

The face images, which are used for training and testing the neural network within the developed recognition system, represent persons of various ethnicities, age and gender. A total of 180 face images of 30 persons with different facial expressions are used, where 90 images are from the ORL face database [13], and more than 90 images are from our own face database. Our face database has been built using face images captured under the following conditions:

- Similar lighting condition
- No physical obstruction
- Head pose is straight without rotation or tilting
- Camera at the same distance from the face

Each person has six different face expressions captured and the image is resized to (100x100) pixels, thus resulting in 90 face images from each face database. Fig. 1 and Fig. 2 show the faces of the 30 persons from our face database and the ORL face database respectively, in addition to examples of the six facial expressions.

A total number of 180 face images of 30 persons with different expressions will be used for the implementation of the proposed intelligent face recognition system. Four out of the six expressions (natural, smiley, sad and surprised) are *approximated* and used for training the neural network within system. The remaining two face expressions are random expressions and will be used together with the *non-approximated* four training face expressions to test the trained neural network.

Averaging is applied only during the neural network training phase where the four facial expressions (natural, smiley, sad and surprised) images are reduced to one face image per person by separately averaging the essential features (eyes, nose, and mouth), thus providing 30 averaged face images for training the neural network. Generalizing or testing the neural network will be implemented using the six facial expressions *without* the averaging process, thus providing 180 face images for testing the trained neural network.

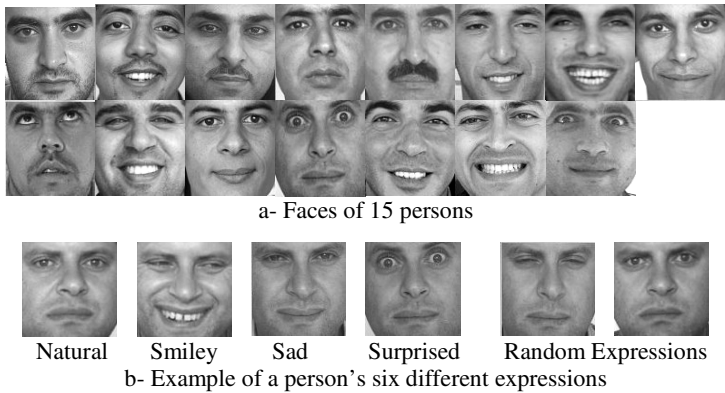


Fig. 1. Faces and expressions from our face database

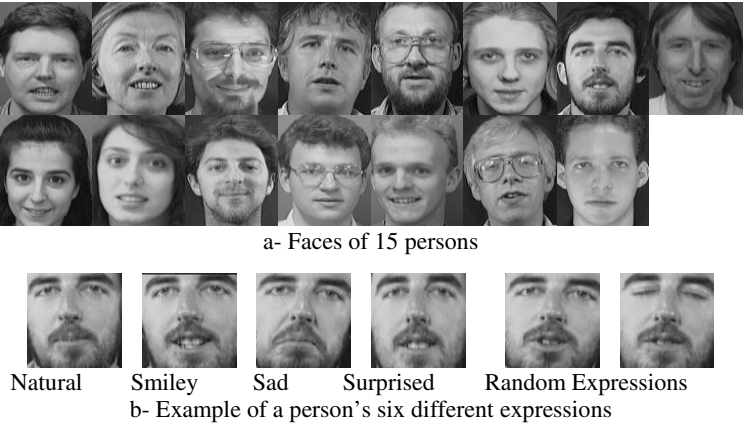


Fig. 2. Faces and expressions from ORL face database [13]

3 Face Recognition System Implementation

The implementation of recognition system comprises the image preprocessing phase and the neural network arbitration phase.

Image preprocessing is required prior to presenting the training or testing images to the neural network. This aims at reducing the computational cost and providing a faster recognition system while presenting the neural network with sufficient data representation of each face to achieve meaningful learning.

The back propagation neural network is trained using approximations of four specific facial expressions for each person, which is achieved by averaging the essential features, and once trained; the neural network is tested or generalized using the six different expressions without approximation.

There are 180 face images of 30 persons with six expressions for each. Training the neural network uses 120 images (which will be averaged to 30 images) representing the 15 persons with four specific expressions. The remaining 60 images of the 15 persons with random different expressions are used together with the 120 training images (prior to averaging) for generalizing the trained neural network, as can be seen in Fig. 3, thus resulting in 180 face images for testing.

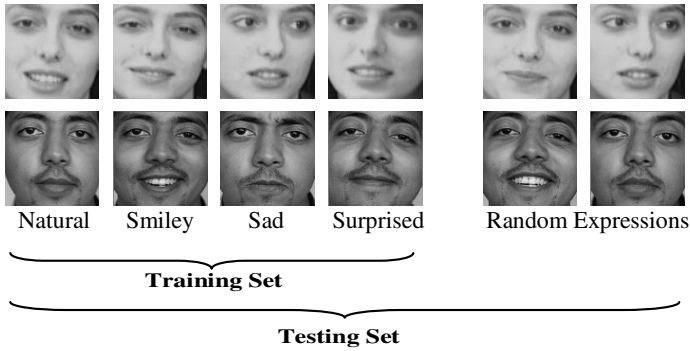


Fig. 3. Examples of training and testing face images

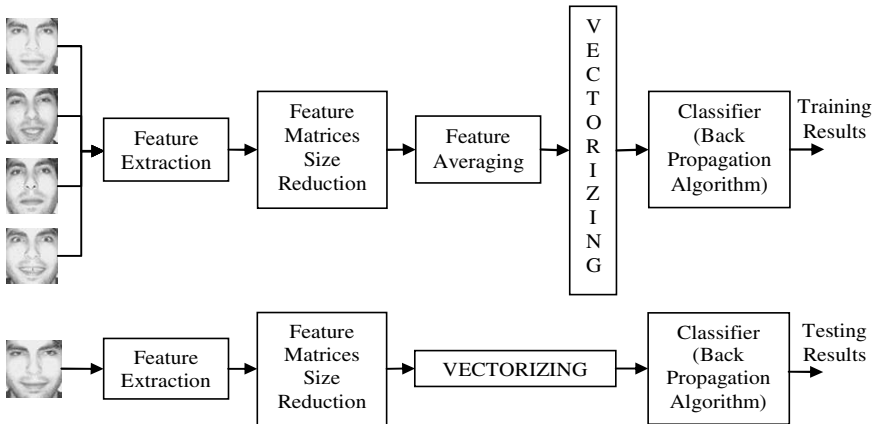


Fig. 4. General architecture of the intelligent face recognition system

3.1 Feature Approximation

The four essential features (eyes, nose and mouth) from four expressions (natural, smiley, sad and surprised) are approximated via averaging into one single vector that represents the person. Approximation is applied *only* prior to training. Fig. 4 shows a block diagram of the recognition system.

The features are, firstly extracted for each facial expression of each subject as shown in Fig. 5. Feature extraction is manually performed using Photoshop, as the work in this paper is not proposing an automatic recognition system, but rather a novel method of training a neural network to recognize faces based on approximated face features. Secondly, the dimensions of each feature are reduced by interpolation. The right eye, left eye, nose and mouth dimensions are reduced to (5 x 10) pixels, (5 x 10) pixels, (7 x 10) pixels and (6 x 17) pixels respectively.

Thus the output matrices dimension after interpolation process will be 1/3 of the input matrices; for example, the 15x30 pixels input matrix will be after interpolation 5x10 pixels. Averaging is then applied where the 120 training images are reduced to 30 averaged images by taking the average for each feature in the four specific expressions for each subject. Feature averaging process for each feature can be implemented using the following equation:

$$f_{avg} = \frac{1}{4} \sum_{i=1}^4 f_i , \tag{1}$$

where f_{avg} is the feature average vector and f_i is feature in expression i of one person. Finally, the averaged features are represented as (272x1) pixel vectors, which will be presented to the input layer of the back propagation neural network.



Fig. 5. Extracted features from different expressions

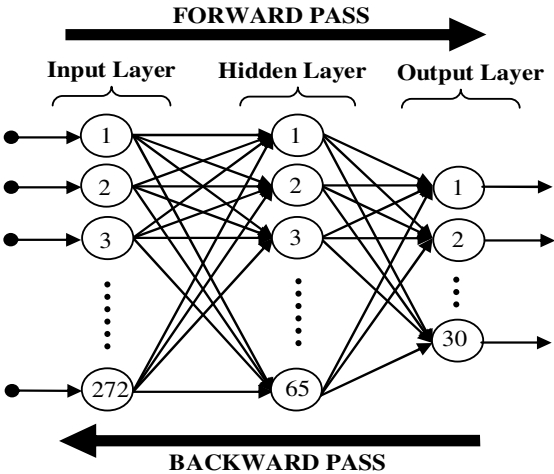


Fig. 6. Back propagation neural network topology

3.2 Neural Network Arbitration

The back propagation algorithm is used for the implementation of the proposed intelligent face recognition system, due to its simplicity and efficiency in solving pattern recognition problems. The neural network comprises an input layer with 272 neurons that carry the values of the averaged features, a hidden layer with 65 neurons and an output layer with 30 neurons which is the number of persons. Fig. 6 shows the back propagation neural network design.

4 Results and Comparison

A fast face recognition system has been developed. The neural network learnt the approximated faces after 3188 iterations and within 265 seconds, whereas the running time for the generalized neural network after training and using one forward pass was 0.032 seconds. These results were obtained using a 1.6 GHz PC with 256 MB of RAM, Windows XP OS and Matlab 6.5 software. Fig. 7 shows the error graph versus iterations during the neural network training. Table 1 shows the final parameters of the successfully trained neural network. The reduction in training and testing time was achieved by the novel method of reducing the face data via averaging selected essential face features for training, while maintaining meaningful learning of the neural network. The face recognition system correctly recognized all averaged face images in the training set as would be expected.

The intelligent system was tested using 180 face images which contain different face expressions that were not exposed to the neural network before; these comprised 90 images from our face database and 90 images from the ORL database. All 90 face images in our database were correctly identified yielding 100% recognition rate with 91.8 % recognition accuracy, whereas, 84 out of the 90 images from the ORL database were correctly identified yielding 93.3% recognition rate with 86.8% recognition accuracy.

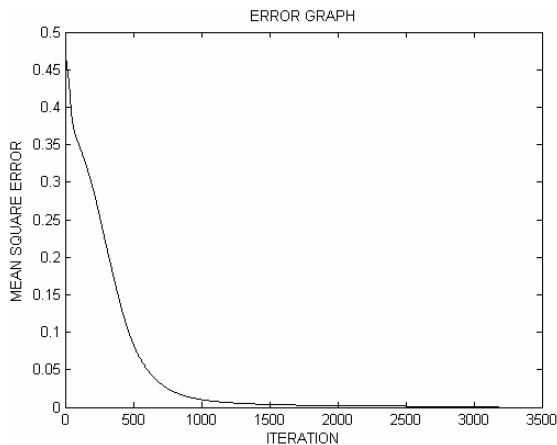


Fig. 7. Error versus iteration graph during network training

Table 1. Neural network final parameters

Number of Input Neurons	272
Number of Hidden Neurons	65
Number of Output Neurons	30
Learning Coefficient	0.0495
Momentum Coefficient	0.41
Minimum Error	0.001
Training Iterations	3188
Training Time	265 Seconds

The total recognition rate for the system was 96.7% where 174 out of the available 180 faces were correctly recognized with an accuracy rate of 89.3%. The recognition rate refers to the percent of correctly recognized faces, whereas the recognition accuracy refers to the classification real output value in comparison to the desired output value of “1”, using binary output coding.

The processing time for face image preprocessing and feature averaging was 7.5 seconds, whereas running the trained neural network took 0.032 seconds. The recognition rates, recognition accuracy and running time of the trained system are shown in Table 2.

Table 2. Recognition Rates, Accuracy and Run Time

Database	Own	ORL	Total
Recognition Rate	100 %	93.3 %	96.7 %
Recognition Accuracy	91.8 %	86.8 %	89.3 %
Recognition Time	0.032 seconds		



Fig. 8. (a) Clear eyeglasses (b) Dark eyeglasses

Further investigations of the capability of the developed face recognition system were also carried out by testing the trained neural network ability to recognize two subjects with eyeglasses. Fig. 8 shows the two persons with and without glasses; person 1 wears clear eyeglasses whereas, person 2 wears darker eyeglasses.

Table 3. Recognition Accuracy With and Without Eyeglasses

Person 1		Person 2	
No Eyeglasses	Clear Eyeglasses	No Eyeglasses	Dark Eyeglasses
96 %	86%	90 %	73 %

The effect of the presence of facial detail such as glasses on recognition performance was investigated. The neural network had not been exposed to the face images with glasses prior to testing. Correct recognition of both persons, with and without their glasses on, was achieved. However, the recognition accuracy was reduced due to the presence of the glasses. The ability of the trained neural network to recognize these faces despite the presence of eyeglasses is due to training the network using feature approximations or “fuzzy” feature vectors rather than using “crisp” feature vectors. Table 3 shows the accuracy rates for both persons with and without glasses.

A comparison has also been drawn between the developed method, which is based on approximating local facial features, and two efficient methods for face recognition; namely, PCA (Eigenfaces) and LDA (Fisherfaces) [14]. The ORL face database has been used for comparison reasons. The recognition rates and accuracies of the face recognition methods using ORL face database are listed in Table 4.

Table 4. Result Comparison of Different Face Recognition Methods

Method	Recognition Rate	Recognition Accuracy
PCA (Eigenfaces)	80%	79.1%
LDA (Fisherfaces)	99.4%	81%
Feature Approximation	93.3%	86.8%

The highest recognition rate obtained using LDA where a recognition rate of (99.4%) was achieved. This was followed by our proposed approximating method (93.3%) and then PCA with an average recognition rate of (80%). On the hand, our method achieved the highest average recognition accuracy (86.6%) in comparison to LDA (81%) and PCA (79.1%). The overall performance of our method has been successful. The robustness, flexibility and speed of this novel face recognition system have been demonstrated through this application.

5 Conclusion

A novel method to intelligent face recognition is proposed in this paper. The method approximates four essential face features (eyes, nose and mouth) from four different facial expressions (natural, smiley, sad and surprised), and trains a neural network using the approximated features to learn the face. Once trained, the neural network

could recognize the faces with different facial expressions. Although the feature pattern values (pixel values) may change with the change of facial expression, the use of averaged-features of a face provides the neural network with an approximated understanding of the identity and is found to be sufficient for training a neural network to recognize that face with any expression, and with the presence of minor obstructions such as eyeglasses.

The successful implementation of the proposed method was shown throughout a real-life implementation using 30 face images showing six different expressions for each. An overall recognition rate of 96.7% with recognition accuracy of 89.3% was achieved.

The use of feature approximation helped reducing the amount of training image data prior to neural network implementation, and provided reduction in computational cost while maintaining sufficient data for meaningful neural network learning. The overall processing times that include image preprocessing and neural network implementation were 272.5 seconds for training and 0.032 seconds for face recognition.

The comparison between the approximation method, LDA and PCA for face recognition shows that our method achieves superior results considering its simplicity and overall high rates of recognition and accuracy.

Further work will include implementing the developed method using different and larger face databases and investigating its efficiency in recognizing partially obstructed faces.

References

1. Brunelli, R., Poggio, T.: Face Recognition: Features versus Templates. *IEEE Trans. PAMI* 15 (1993) 1042-1052
2. Turk, M., Pentland, A.: Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, Vol. 3 (1991) 72-86
3. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: Face Recognition using class specific linear projection. *Proc. ECCV*, (1996) 45-58
4. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear Image Analysis for Facial Recognition. *Proceedings of International Conference on Pattern Recognition* (2002)
5. Savvides, M., Vijaya Kumar, B.V.K., Khosla, P.K.: Eigenphases vs. Eigenfaces. *17th International Conference on Pattern Recognition*, Vol. 3 (2004) 810-813
6. Shen, L., Bai, L.: Gabor Wavelets and Kernel Direct Discriminant Analysis for Face Recognition. *17th International Conference on Pattern Recognition*, Vol. 1 (2004) 284-287
7. Dai, G., Qian, Y., Jia, S.: A Kernel Fractional-Step Nonlinear Discriminant Analysis for Pattern Recognition. *17th International Conference on Pattern Recognition*, Vol. 2 (2004) 431-434
8. Zhao, S., Grigat, R.: Multiblock-Fusion Scheme for Face Recognition. *17th International Conference on Pattern Recognition*, Vol. 1 (2004) 309-312
9. Ahonen, T., Pietikainen, M., Hadid, A.: Face Recognition Based on the Appearance of Local Regions. *17th International Conference on Pattern Recognition*, Vol. 3 (2004) 153-156
10. Zhang, B., Zhang, H., Ge, S.: Face recognition by applying wavelet subband representation and kernel associative memory. *IEEE Transactions on Neural Networks*, Vol. 15 (2004) 166-177

11. Fan, X., Verma, B.: A Comparative Experimental Analysis of Separate and Combined Facial Features for GA-ANN based Technique. Proceedings of International Conference on Computational Intelligence and Multimedia Applications (2005) 279-284
12. Khashman, A.: Face Recognition Using Neural Networks and Pattern Averaging. Lecture Notes on Computer Science, Vol. 3972. Springer-Verlag, Berlin Heidelberg New York (2006) 98–103
13. Cambridge University, Olivetti Research Laboratory face database. <http://www.uk.research.att.com/facedatabase.html>
14. Lu, X., Wang, Y., Jain, A.K.: Combining Classifiers for Face Recognition. IEEE International Conference on Multimedia & Expo (ICME'03), Vol. III (2003) 13-16

A Methodology for Building Semantic Web Mining Systems

Francesca A. Lisi

Dipartimento di Informatica, Università degli Studi di Bari,
Via E. Orabona 4, I-70125 Bari, Italy
`lisi@di.uniba.it`

Abstract. In this paper we present a methodology based on interoperability for building Semantic Web Mining systems. In particular we consider the still poorly investigated case of mining the Semantic Web layers of ontologies and rules. We argue that Inductive Logic Programming systems could serve the purpose if they were more compliant with the standards of representation for ontologies and rules in the Semantic Web and/or interoperable with well-established Ontological Engineering tools that support these standards.

1 Introduction

Semantic Web Mining [16] is a new application area which aims at combining the two areas of Semantic Web [2] and Web Mining [9] from a twofold perspective. On one hand, the new semantic structures in the Web can be exploited to improve the results of Web Mining. On the other hand, the results of Web Mining can be used for building the Semantic Web. Mining the Semantic Web layers of ontologies and rules still remains poorly investigated. Yet systems based on Inductive Logic Programming (ILP) [13] could serve the purpose if they were more compliant with the standards of representation for ontologies and rules in the Semantic Web¹ and/or interoperable with well-established tools for Ontological Engineering (OE) [4], e.g. Protégé-2000 [14], that support these standards.

In [11] we have studied the applicability of the ILP system $\mathcal{AL}$ -QUIN in Semantic Web Mining. We remind that $\mathcal{AL}$ -QUIN relies on the expressive and deductive power of the hybrid knowledge representation and reasoning system $\mathcal{AL}$ -log [3] and supports a variant of the task of frequent pattern discovery [12]. In this paper we present a middleware, *SWING*, that integrates $\mathcal{AL}$ -QUIN and Protégé-2000 in order to enable Semantic Web Mining applications of $\mathcal{AL}$ -QUIN. This solution suggests a methodology for building Semantic Web Mining systems, i.e. the cooperation of existing systems on the basis of interoperability.

¹ Whereas the mark-up language OWL (<http://www.w3.org/2004/OWL/>) for ontologies is already undergoing the standardization process at W3C, the mark-up language SWRL (<http://www.w3.org/Submission/SWRL/>) for rules is just a proposal that has been very recently submitted to W3C.

2 The Middleware *SWing*

As illustrated in Figure 1, *SWing* interoperates via API with the OWL Plugin for Protégé-2000 to benefit from its facilities for browsing and reasoning on OWL ontologies. Note that, since the design of OWL has been based on Description Logics (DLs) [1] (more precisely on the DL *SHIQ* [6]), the OWL Plugin has been conceived so that it can directly access DL reasoners such as RACER [5].

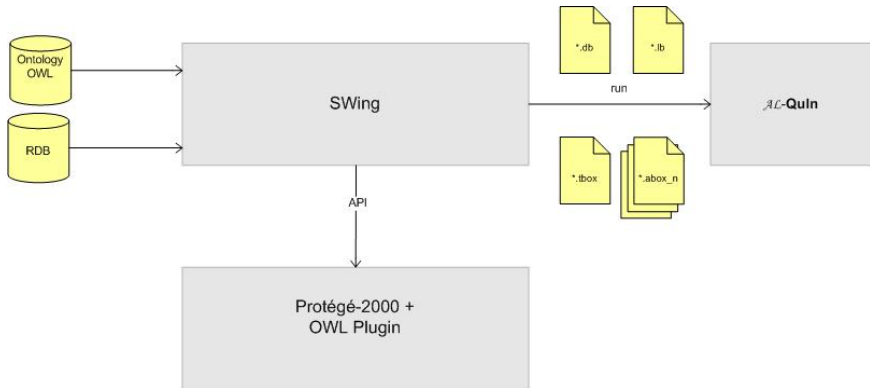


Fig. 1. Architecture and I/O of *SWing*

The software component *SWing* assists users of *AL-QUIN* in the design of Semantic Web Mining sessions. A wizard provides guidance for the selection of the (hybrid) data set to be mined, the selection of the reference concept and the task-relevant concepts (see Figure 2), the selection of the relations - among the ones appearing in the relational component of the data set chosen or derived from them - with which the task-relevant concepts can be linked to the reference concept in the patterns to be discovered, the setting of support thresholds for each level of description granularity and of several other parameters required by *AL-QUIN*. These user preferences are collected in a file (see output file **.lb* in Figure 1) that is shown in preview to the user at the end of the assisted procedure for confirmation.

Example 1. To illustrate *SWing* we refer to a Semantic Web Mining session for the task of finding frequent patterns in the on-line CIA World Fact Book² (data set) that describe Middle East countries (reference concept) w.r.t. the religions believed and the languages spoken (task-relevant concepts) at three levels of granularity ($maxG = 3$). Further details can be found in [11]. The input for this Semantic Web Mining session is a knowledge base $\mathcal{B}_{CIA}$ that integrates an OWL ontology (file *cia_exp1.owl*) with a DATALOG database (file *cia_exp1.edb*)

² <http://www.odci.gov/cia/publications/factbook/>

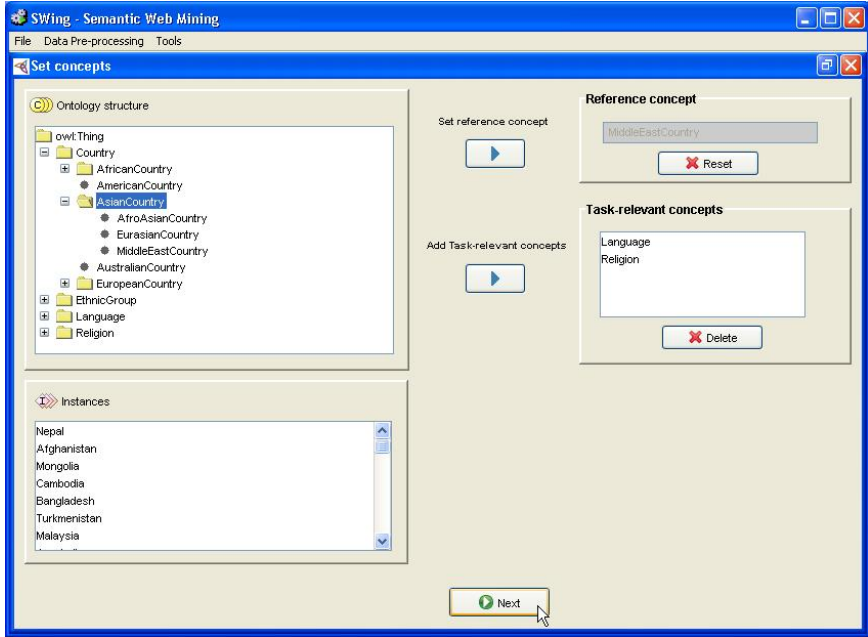


Fig. 2. Selecting concepts from the ontology with *SWING*

containing facts³ extracted from the on-line 1996 CIA World Fact Book. The output DATALOG database `cia_exp1.db` enriches `cia_exp1.edb` with clauses such as:

```
speaks(Code, Lang) ← language(Code, Lang, Perc),
                    c.Country(Code), c.Language(Lang).
```

that define views on the relations already occurring in `cia_exp1.edb`. The editing of the views is done visually by means of the form shown in Figure 3.

We shall focus now on the step of concept selection because it exploits the services offered by Protégé-2000 more deeply. To this aim we would like to point out that the internal representation language in $\mathcal{AL}$ -QUIN is a kind of DATALOG^{OI} [15], i.e. the subset of DATALOG[≠] equipped with an equational theory that consists of the axioms of Clark's Equality Theory augmented with one rewriting rule that adds *inequality atoms* $s \neq t$ to any $P \in \mathcal{L}$ for each pair (s, t) of distinct terms occurring in P . Note that concept assertions are rendered as *membership atoms*, e.g. $'IR':\text{MiddleEastCountry}$ becomes $\text{c.MiddleEastCountry}('IR')$. Therefore the step of concept selection in *SWING* also triggers some supplementary computation aimed at making a OWL background knowledge Σ usable by $\mathcal{AL}$ -QUIN. To achieve this goal, it supplies the following functionalities:

³ <http://www.dbis.informatik.uni-goettingen.de/Mondial/mondial-rel-facts.flp>

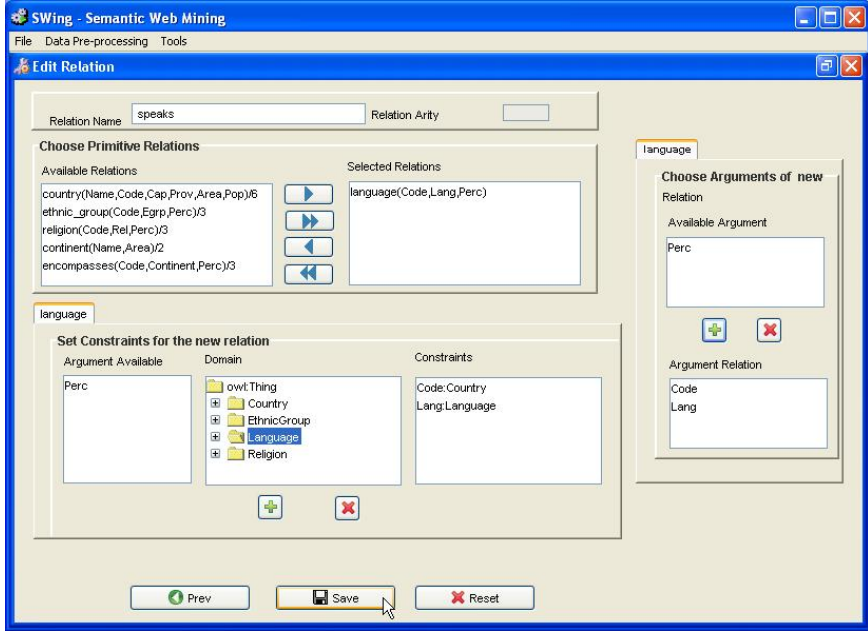


Fig. 3. Editing of derived relations with SWING

- levelwise retrieval w.r.t. Σ
- translation of both (asserted and derived) concept assertions and subsumption axioms of Σ to DATALOG^{OI} facts

The latter relies on the former, meaning that the results of the levelwise retrieval are exported to DATALOG^{OI} (see output files **.abox_n* and **.tbox* in Figure 1). The *retrieval* problem is known in DLs literature as the problem of retrieving all the individuals of a concept C [1]. Here, the retrieval is called *levelwise* because it follows the layering of the TBox $\mathcal{T}$: individuals of concepts belonging to the l -th layer $\mathcal{T}^l$ of $\mathcal{T}$ are retrieved all together.

Example 2. The DATALOG^{OI} rewriting of the concept assertions derived for $\mathcal{T}^2$ produces facts like:

```
c_AfroAsiaticLanguage('Arabic').
c_IndoEuropeanLanguage('Persian').
c_UralAltaicLanguage('Kazak').
c_MonotheisticReligion('ShiaMuslim').
c_PolytheisticReligion('Druze').
```

that are stored in the file *cia_exp1.abox_2*.

The file *cia_exp1.tbox* contains a DATALOG^{OI} rewriting of the taxonomic relations of $\mathcal{T}$ such as:


```

hierarchy(c_Language,1,null,[c_Language]).
hierarchy(c_Religion,1,null,[c_Religion]).

```

for the layer $\mathcal{T}^1$ and

```

hierarchy(c_Language,2,c_Language,
          [c_AfroAsiaticLanguage, c_IndoEuropeanLanguage, ...]).
hierarchy(c_Religion,2,c_Religion,
          [c_MonotheisticReligion, c_PolytheisticReligion]).

```

for the layer $\mathcal{T}^2$ and

```

hierarchy(c_Language,3,c_AfroAsiaticLanguage,[c_AfroAsiaticLanguage]).
...
hierarchy(c_Language,3,c_IndoEuropeanLanguage,
          [c_IndoIranianLanguage, c_SlavicLanguage]).
hierarchy(c_Language,3,c_UralAltaicLanguage,[c_TurkicLanguage]).
hierarchy(c_Religion,3,c_MonotheisticReligion,
          [c_ChristianReligion, c_JewishReligion, c_MuslimReligion]).

```

for the layer $\mathcal{T}^3$.

Note that the translation from OWL to DATALOG^{OI} is possible because we assume that *all* the concepts are named. This means that an equivalence axiom is required for each complex concept in the knowledge base. Equivalence axioms help keeping concept names (used within constrained DATALOG clauses) independent from concept definitions.

3 Conclusions and Future Work

The middleware *SWING* implements a methodology for building Semantic Web Mining systems that follows engineering principles. Indeed it promotes the reuse of existing systems (*AL-QUIN* and *Protégé-2000*) and the adherence to standards (either normative - see OWL for the Semantic Web - or *de facto* - see Prolog for ILP). Furthermore the resulting artifact overcomes the capabilities of the two systems when considered stand-alone. In particular, *AL-QUIN* was originally conceived to deal with *ALC* ontologies. Since *ALC* is a fragment of *SHIQ*, *SWING* allows *AL-QUIN* to deal with more expressive ontological background knowledge by supplying a facility for compiling OWL down to DATALOG^{OI}. In this respect, the pre-processing method proposed by Kietz [8] to enable legacy ILP systems to work within the framework of the hybrid KR&R system *CARIN* [10] is related to ours but it lacks an application. Analogously, the method proposed in [7] for translating OWL to disjunctive DATALOG is far too general with respect to the specific needs of our application.

For the future we plan to extend *SWING* with facilities for extracting information from semantic portals and for presenting patterns generated by *AL-QUIN*.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
2. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May, 2001.
3. F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. $\mathcal{AL}$ -log: Integrating Datalog and Description Logics. *Journal of Intelligent Information Systems*, 10(3):227–252, 1998.
4. A. Gómez-Pérez, M. Fernández-López, and O. Corcho. *Ontological Engineering*. Springer, 2004.
5. V. Haarslev and R. Möller. Description of the RACER System and its Applications. In C.A. Goble, D.L. McGuinness, R. Möller, and P.F. Patel-Schneider, editors, *Working Notes of the 2001 International Description Logics Workshop (DL-2001)*, volume 49 of *CEUR Workshop Proceedings*, 2001.
6. I. Horrocks, P.F. Patel-Schneider, and F. van Harmelen. From *SHIQ* and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.
7. U. Hustadt, B. Motik, and U. Sattler. Reducing *SHIQ*-description logic to disjunctive datalog programs. In D. Dubois, C.A. Welty, and M.-A. Williams, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, pages 152–162. AAAI Press, 2004.
8. J.-U. Kietz. Learnability of description logic programs. In S. Matwin and C. Sammut, editors, *Inductive Logic Programming*, volume 2583 of *Lecture Notes in Artificial Intelligence*, pages 117–132. Springer, 2003.
9. R. Kosala and H. Blockeel. Web Mining Research: A Survey. *SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, ACM*, 2, 2000.
10. A.Y. Levy and M.-C. Rousset. Combining Horn rules and description logics in CARIN. *Artificial Intelligence*, 104:165–209, 1998.
11. F.A. Lisi and F. Esposito. Mining the Semantic Web: A Logic-Based Methodology. In M.-S. Hacid, N.V. Murray, Z.W. Ras, and S. Tsumoto, editors, *Foundations of Intelligent Systems*, volume 3488 of *Lecture Notes in Artificial Intelligence*, pages 102–111. Springer, 2005.
12. H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.
13. S.-H. Nienhuys-Cheng and R. de Wolf. *Foundations of Inductive Logic Programming*, volume 1228 of *Lecture Notes in Artificial Intelligence*. Springer, 1997.
14. N. Fridman Noy, R.W. Fergerson, and M.A. Musen. The Knowledge Model of Protégé-2000: Combining Interoperability and Flexibility. In R. Dieng and O. Corby, editors, *Knowledge Acquisition, Modeling and Management*, volume 1937 of *Lecture Notes in Computer Science*, pages 17–32. Springer, 2000.
15. G. Semeraro, F. Esposito, D. Malerba, N. Fanizzi, and S. Ferilli. A logic framework for the incremental inductive synthesis of Datalog theories. In N.E. Fuchs, editor, *Proceedings of 7th International Workshop on Logic Program Synthesis and Transformation*, volume 1463 of *Lecture Notes in Computer Science*, pages 300–321. Springer, 1998.
16. G. Stumme, A. Hotho, and B. Berendt. Semantic Web Mining: State of the art and future directions. *Journal of Web Semantics*, 4(2):124–143, 2006.

Content-Based Image Filtering for Recommendation

Kyung-Yong Jung

School of Computer Information Engineering,
Sangji University, Korea
kyjung@sangji.ac.kr

Abstract. Content-based filtering can reflect content information, and provide recommendations by comparing various feature based information regarding an item. However, this method suffers from the shortcomings of superficial content analysis, the special recommendation trend, and varying accuracy of predictions, which relies on the learning method. In order to resolve these problems, this paper presents content-based image filtering, seamlessly combining content-based filtering and image-based filtering for recommendation. Filtering techniques are combined in a weighted mix, in order to achieve excellent results. In order to evaluate the performance of the proposed method, this study uses the EachMovie dataset, and is compared with the performance of previous recommendation studies. The results have demonstrated that the proposed method significantly outperforms previous methods.

1 Introduction

With the recent advancement in information technology, the amount of information that users can access has increased exponentially. Due to the development of the Internet, a wide variety of information is being produced and distributed rapidly in digital form. In this excess of information, it is not straightforward for users to search and retrieve desired information in a short time period. Accordingly, the personalized recommendation system, in which users can utilize and control filtered information efficiently, has appeared in electronic commerce. It applies to information filtering for building a system that can predict and recommend customized items to users. For instance, a recommendation system on Amazon.com, CDNow.com, and Levis.com suggests items to users based on in other items users have registered interest [5][12].

In this paper, content-based image filtering aims to study how content-based filtering and image-based filtering can be combined, in order to provide improved performance for a user-adapting recommendation system. Content-based filtering can reflect content information, and provides recommendations by comparing feature information regarding an item, and the profile of user preferences. However, this suffers from the shortcomings of superficial content analysis, the special recommendation trend, and the varying accuracy of prediction depending on the learning method. Therefore, the purpose of this paper is to solve the problems of content-based filtering. This paper suggests content-based image filtering, seamlessly combining content-based filtering and image-based filtering for recommendation.

2 System Overview

Content-based image filtering is a project that aims to study how the combination of content-based filtering and image-based filtering can be used to build a user-adapting recommendation system. Figure 1 presents a system overview for content-based image filtering. This is composed of content-based filtering and image-based filtering.

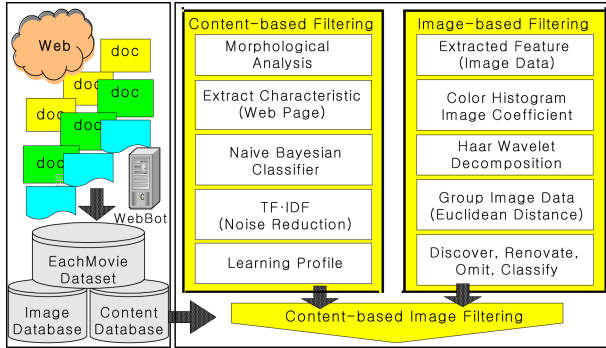


Fig. 1. System overview for content-based image filtering

The content information database presents web pages collected by WebBot [3] as a single-word position vector using morpheme analysis, instead of single words. The WebBot uses URLs provided in the EachMovie dataset to download movie content from IMDb(<http://www.imdb.com>). Since learning the profile through use of the simple Naïve Bayesian classifier, all words appearing in web pages are extracted. It is difficult to accurately reflect the true characteristics of web pages. Therefore, content-based filtering extracts characteristics of web pages through use of TF-IDF from web pages. It also provides weight to characteristics extracted from web pages, so incorrect classification resulting from noise is reduced greater than the simple Naïve Bayesian classifier. This research uses image-based filtering, which extracts the feature from image data a user is interested in, in order to improve the superficial problem of content analysis. Image data is discovered, renovated, omitted, and classified. This uses content-based image filtering through combining content-based filtering with the technique of learning the profile and using image-based filtering in order to reinforce the special trend that recommendation provides similar items.

3 Content-Based Image Filtering

3.1 Extraction Information

The current prototype system, WebBot: Web Robot Agent [3][4] uses a content information database of movie content information extracted from web pages in the IMDb. Therefore, the system's current content information regarding titles consists of textual meta-data rather than the actual text of the items themselves. The IMDb

subject search is performed in order to obtain a list of movie-description URLs of broadly relevant titles. The WebBot then downloads each of these pages and uses a simple pattern based information extraction system to extract data associated with each title. Figure 2 presents WebBot for extracting information, and an example web page. Information extraction is the task of locating specific pieces of information from a document, thereby obtaining useful structured data from unstructured text. A WebBot follows the IMDb link provided for every movie in the EachMovie dataset [8] and collects information from the various links off the main URL. The content information of every movie is represented as a set of features. Each feature is represented simply as a group of words. The IMDb produces information regarding related directors and movie titles. However, WebBot treats this information as additional content associated with the movie. The text in each feature is then processed into an unordered group of words, and the examples represented as a vector of groups of words.

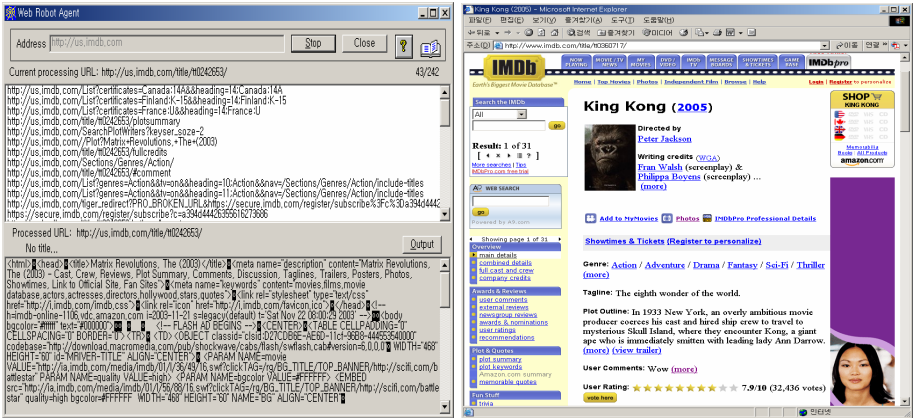


Fig. 2. WebBot: Web Robot Agent, example web page

3.2 Content-Based Filtering

We use a multinomial text model, in which a web page is modeled as an ordered sequence of ordered sequence of word events drawn from the same vocabulary, V . The Naïve Bayesian assumption states that the probability of each word is dependent on the web page classes but independent of the word's context and position [10]. For each class c_j and word, $w_k \in V$, the probability, $P(c_j)$ and $P(w_k | c_j)$ must be estimated from the training data. Then the posterior probability of each class given a web page, D , is computed using Bayes rule by Equation (1).

$$p(c_j | D) = \frac{P(c_j)}{P(D)} \prod_{i=1}^{|D|} P(a_i | c_j) \quad (1)$$

Where a_i is the i th word in the web page, and $|D|$ is the length of the page in words. Since for any given page, the prior $P(D)$ is a constant, this factor can be ignored if all that is desired is a rating rather than a probability estimate. A ranking is produced by

sorting pages by their odds ratio, $P(c_1|D)/P(c_0|D)$, where c_1 represents the positive class and c_0 represents the negative class [9]. An item is classified as positive if the odds are greater than 1, and negative otherwise. In case, since items are represented as a vector of web pages, d_m , one for each slot s_m , the probability of each word given the class and the slot $P(w_k|c_j, s_m)$, must be estimated and the posterior class probability for a film, F , computed using Equation (2).

$$p(c_j | F) = \frac{P(c_j)}{P(F)} \prod_{m=1}^S \prod_{i=1}^{|d_m|} f_{nk} \cdot [\text{Log}_2 \frac{n}{DF} + 1] \cdot P(a_{m,i} | c_j, s_m) \quad (2)$$

Where S is the number of slots and $a_{m,i}$ is the i th word in the m th slot. The class with the highest probability determines the predicted rating. The *Laplace* smoothing [10] is used to avoid zero probability estimates for words that do not appear in the limited training set. Finally, calculation with logarithms of probabilities is used to avoid underflow. Naïve Bayesian classifier through use of TF-IDF [9][10] makes morphological analysis to extract characteristic of web pages and extracts only nouns from its outcome. f_{nk} is relative frequency of word n_k against all words within all web pages and n is the number of web pages and DF is the number of learning web pages where word n_k appeared. If characteristic of web pages is $\{a_1, a_2, \dots, a_i, \dots, a_D\}$, Naïve Bayesian classifier classifies web pages into one class among $\{c_1, c_2, \dots, c_i, \dots, c_j\}$. Equation (3) is used to give probability of word $(w_{k1}, w_{k2}, \dots, w_{k(r-1)}, w_{kr})$ within c_j, s_m is expressed as $p((w_{k1}, w_{k2}, \dots, w_{k(r-1)}, w_{kr}) | c_j, s_m)$.

$$p((w_{k1}, w_{k2}, w_{k(r-1)}, w_{kr}) | c_j, s_m) = \sum_{e=1}^N f_{nk} \cdot [\text{Log}_2 \frac{n}{DF} + 1] \cdot a_{ej} / (\sum_{e=1}^N a_{ej} | d_m) \quad (3)$$

When predicting for a target item, content-based filtering computes the posterior probability of each class given in the target item. Then the predicted rating of the posterior probability distribution for the categories is computed and used as predicted ratings by Equation (4). Here, $P(i)$ is the posterior probability for class c_j . It treats items rated 1-3 as negative instances, and those rated 4-6 as positive instances. The scores are ranked based on the natural log of the posterior odds of positive [4].

$$P^{con}(u, i) = \sum_{i=1}^6 i \cdot P(i) \quad (4)$$

The predicted ratings are used, rather than choosing the most probable class better representing the continuity of the ratings.

3.3 Image-Based Filtering

Image-based filtering assigns the original image data to a relatively small number of groups, where each group represents a set of pixels coherent in their color and local image data properties; the motivation is to reduce the amount of raw data represented, while preserving the content information required for the image data understanding task. It is reasonable to expect that the image data with similar content information will be almost equally interesting to users [6].

Two feature extraction components were implemented; color histograms and image data coefficients. Color is an important cue in extracting information from the image data. Color histograms are commonly used image-analysis tools and have

proven to be useful. The original image data is available in RGB format, where each pixel is defined by the value of the three components red, blue, and green. The hue-saturation-value color space is treated as a cone: for a given point (h, s, v) , the human perception of colors is modeled more accurately. The HSV coefficients are quantized to yield 166 different colors. h and sv are the angular and radial coordinates of the point on a disk of radius v and height v ; all coordinates range from 0 to 1. Points with a small v are black, regardless of their h and s values. The cone representation maps of all such points in the apex of the cone, are close to one another. For the image data, the histogram of these 166 colors is computed. This encoding allows internalization of the fact that hue differences are meaningless for small saturation. However, it ignores the fact that for large values and saturations, hue differences are perceptually more relevant than saturation and value differences [4][6]. In order to compare two image data, the Euclidean distance between their color histograms is computed.

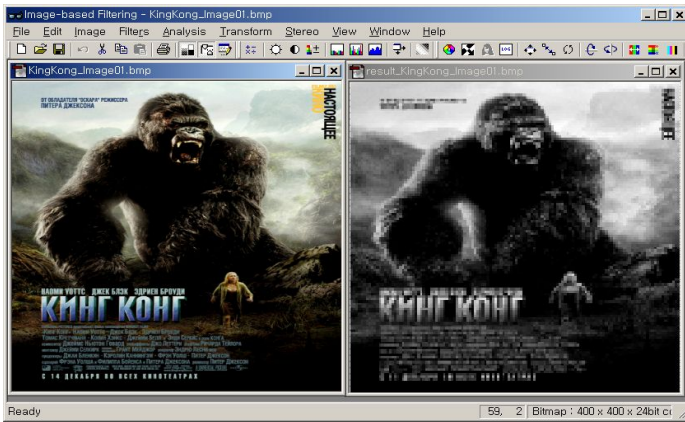


Fig. 3. Original image data / Wavelet decomposition (Haar 2 Wavelet: elapsed 0.048 sec)

The image data is quantized as binary values, and each pixel of the original image data is associated with a binary vector of length 9. The histogram is the feature vector associated with analysis of the image data. As presented in Figure 3, in order to determine the similarity between the image data, all image data are decomposed according to wavelet decomposition using the 2-dimensional Haar transform. From the decomposition, a feature histogram is derived, and can then be compared by the use of vector metric. A linear estimate is used for image-based filtering, which is illustrated in the following Equation (5). $P^{image}(u, i)$ represents image-based filtering of image data i for user u .

$$P^{image}(u, i) = \sum_{j=1}^{imagenum} \left(\sum_{a \in Group_j(i)} r_{u,a} / |Group_j(i)| \right) \quad (5)$$

If a prediction is to be made for a user u and image data i , all the image data previously rated by user u are grouped into $Group_j(i)$ according to distance to the target image data i . The distance is then sorted and grouped.

3.4 Content-Based Image Filtering

For the following section, it is assumed that content-based image filtering, is considered as a personalized item recommendation system. Content-based image filtering is rather an extension of content-based filtering and image-based filtering. As research leads to additional image-analysis tools, the extension approach should not limit the number of content-based filtering extensions. Therefore, content-based image filtering $p^{con-image}(u,i)$ using Equation (6) is used.

$$p^{con-image}(u,i) = weight^{con} p^{con}(u,i) + weight^{image} p^{image}(u,i) \quad (6)$$

In the above equation, $p^{con}(u,i)$ corresponds to content-based filtering, as presented in Equation (4). $p^{image}(u,i)$ corresponds to image-based filtering, as showed in Equation (5). The denominator is a normalization factor that ensures all weights sum to one. The weights ($weight^{con}$, $weight^{image}$) are estimated by the use of linear regression with a set-aside subset of the rating, so that the weights are adapted to the relevance feedback [10]. The estimation of the weights should be repeated in the content information database. Content-based image filtering approaches use parameters to control content-based filtering, and image-based filtering. The parameters determine the weight automatically as the sum of each particular filtering [5].

The criteria of recall and accuracy of the predictive result has been evaluated through the EachMovie dataset. For evaluation, the web pages and the image data related to the movie have been extracted from WebBot [3]. This information is extracted to create 1,000 web pages and 1,000 image data. The recall and accuracy relates to the change in the image-based weight and the content-based weight investigated. Figure 4 indicates the change in accuracy and recall according to the image-based weight.

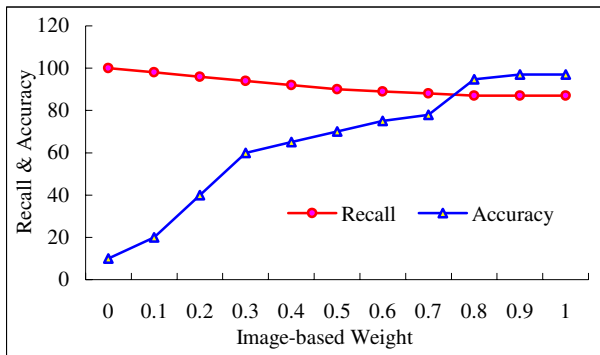


Fig. 4. The change in recall and accuracy according to vary the image-based weight

Figure 4 demonstrates that the greater the image-based weight, the more accurate prediction becomes, but the lower recall becomes. However, recall is almost consistent and accuracy recorded high is not less than 0.76 of the image-based weight.

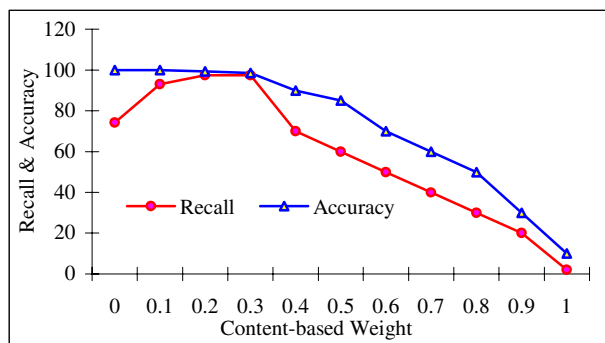


Fig. 5. The change in recall and accuracy depending on the varying content-based weight

Accordingly, in order to provide the most proper image-based weight, the image-based weight should be fixed at not less than 0.76. Figure 5 represents a change in recall, and accuracy, depending on the varying content-based weight. The criteria of evaluating recall and accuracy are the same as the image-based weight.

The curve of accuracy and recall is identical to the content-based weight of 0.29 and at this point, the most proper content-based weight is extracted. However, if the content-based weight is not less than 0.29, both recall and accuracy become lower. Accordingly, in order to extract the most trustworthy weight, a content-based weight not more than 0.29 should be designated. Therefore, it is found through experiment that the following values for these parameters: $weight^{image}=0.76$, $weight^{con}=0.24$.

After reviewing the recommendation, the user may assign their own rating to items they believe to be incorrectly ranked, and retrain the system to produce improved recommendations. As with relevance feedback in information retrieval, this cycle can be repeated several times to produce optimum performance. In addition, as new items are provided, content-based image filtering can track any change in user preferences and alter its recommendation.

4 Evaluation

In this section, the experimental methodology and metrics used to compare different prediction algorithms are described; and the results of this experiment are presented. The user-movie ratings provided by the EachMovie dataset [8] and the movie details from the Internet Movie Database (IMDb) are used. The EachMovie dataset was obtained by the DEC Systems Research Center, by running a movie recommendation service for 18 months. The data set contains 72,916 users who entered a total of 2,811,983 numeric ratings for 1,628 different movies. For the evaluation, the EachMovie dataset was processed for a total of 20,864 users, and each user rated at least 80 movies and 1,612 kinds of movies, through data integrity. This constituted the training set for the Naïve Bayesian learning phase. 20,862 users were selected for the systematic sample, based on the preprocessed EachMovie dataset. Each rating comes from the set $\{0.0, 0.2, 0.4, 0.6, 0.8, \text{ and } 1.0\}$ where 0.0 was the lowest score and 1.0

was the highest. The movies were categorized into 10 categories: action, animation, art foreign, classic, comedy, drama, family, horror, romance, and thriller, where a movie may belong to more than one class.

In this paper, Rank Score Measure (RSM) and Mean Absolute Error (MAE) are used to gauge performance. RSM is used to evaluate the performance of systems that recommend items from ranked lists. MAE is used, in order to evaluate single item recommendation systems [2][5].

The RSM of an item in a ranked list is determined by user evaluation or user visits. The RSM is measured under the premise that the probability of choosing an item lower in the list decreases exponentially. Suppose that each item is put in a decreasing order of value j , based on the weight of the preference. Equation (7) calculates the expected utility of user's RSM using the ranked item list.

$$R_a = \sum_j \frac{\max(V_{a,j} - d, 0)}{2^{(j-1)/(\alpha-1)}} \quad (7)$$

In Equation (7), d is the mid-average value of the item, and α is the half-life. The half-life is the number of items in a list that have a 50/50 chance of either review or visit. In the evaluation phase of this paper a half-life value of 5 is used. In Equation (8), the RSM is used to measure the accuracy of the predictions regarding the user.

$$R = 100 \times \frac{\sum_a R_a}{\sum_a R_a(\max)} \quad (8)$$

In Equation (8), if the user has evaluated or visited an item ranking highly in a ranked list, $R_u(\max)$ is the maximum expected utility of the RSM.

The accuracy of the MAE, expressed as Equation (9), is determined by the absolute value of the difference between the predicted value and real value of user evaluation. $P_{a,j}$ is the predicted preference, $v_{a,j}$ the real preference, and m_a the number of items that have been evaluated by the new user.

$$s_a = \frac{1}{m_a} \sum_{j \in p_a} |p_{a,j} - v_{a,j}| \quad (9)$$

In order to verify this hypothesis, the following experiments were conducted. The computer for the experiment consisted of a PentiumIV, 2-way CPU, 3.0GHz, 512 MB RAM PC. The algorithms used MS-Visual Studio C++ 6.0, and MS-SQL Server 2000. This paper uses the following methods: The proposed content-based image filtering (CbIF), and the former memory based methods used a Pearson correlation coefficient (P_Corr), the recommendation method only used content-based filtering (Content). Predictions were computed for the items using each of the different predictors. The quality of the various prediction algorithms was measured by comparing the predicted values for the ratings to the actual ratings. Various methods were used to compare performance by changing the number of clustering users.

Figure 6 and Figure 7 demonstrate the RSM and MAE of the number of users based on Equation (8) and Equation (9). In Figure 6 and Figure 7, as the number of users increases, the performance of the CbIF and the P_Corr also increases, whereas the method using the Content presents no notable change in performance. In terms of prediction accuracy, it is evident that the CbIF, which uses content-based image filtering for recommendation, is superior to the P_Corr.

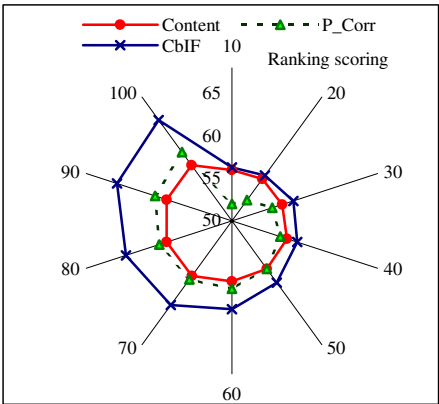


Fig. 6. RSM at varying the number of users

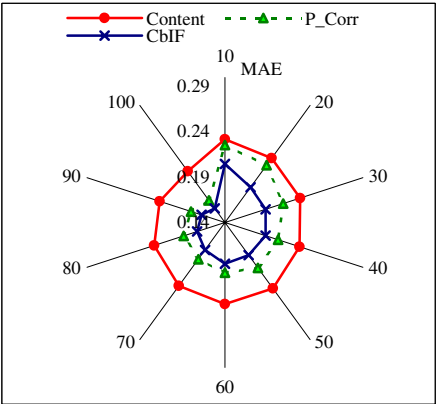


Fig. 7. MAE at varying the number of users

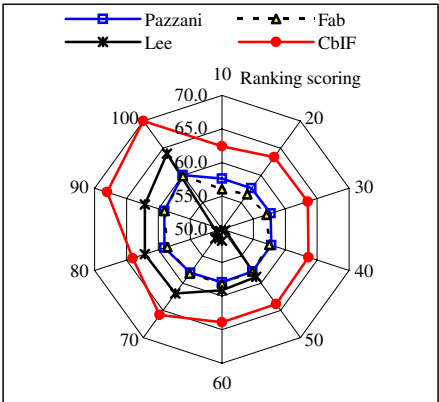


Fig. 8. Rank scoring of n th rating

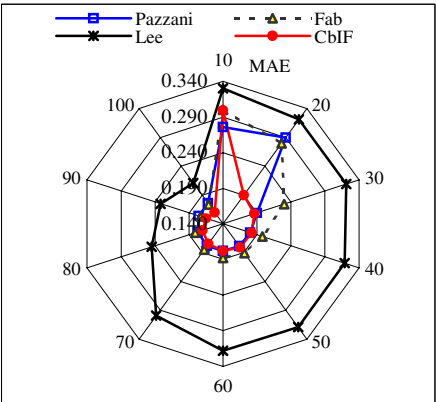


Fig. 9. MAE of n th rating

The proposed recommendation system is considered using content-based image filtering (CbIF) and hypothesized that this system would outperform the stand-alone content-based filtering approach. However, it is also important to compare the proposed approach with that obtained using a combination of content-based filtering and image-based filtering. In comparing the proposed method and those proposed by Lee [7], Pazzani [11], and Fab [1] analysis of the predictive accuracy values, such as RSM and MAE, can be achieved. Figure 8 and Figure 9 demonstrate RSM and MAE as the frequency, which is increased, and where the user evaluates the n th rating number. Figure 8 and Figure 9 demonstrate a system proposed by Lee [7], solving the varying accuracy of prediction depending on the learning method, exhibiting lower performance when the frequency of evaluations is lower. The other methods demonstrate higher performance than that of Lee. As a result, the method developed by Pazzani and CbIF, solving both the superficial content analysis and the special recommendation trend, present the highest accuracy rates.

5 Conclusion

In this paper, content-based image filtering for recommendation is proposed. The shortcomings of content-based filtering are identified, and the method of solving these problems through combining content-based filtering and image-based filtering, is presented. It is believed that the contributions of this paper can be separated into two areas. First, each filtering technique is combined in a weighted mix to achieved good results. It is shown that a combination of content-based filtering and image-based filtering is able to extract a content-based weight of 0.76 and an image-based weight of 0.24. Second, it is shown that content-based image filtering performs significantly better than content-based filtering, and image-based filtering. In the future, if the proposed method is to be used with a memory based recommendation system, it is expected that accuracy will be enhanced.

References

1. Balabanovic, M., Shoham, Y.: Fab: Content-based, Collaborative Recommendation. Communication of the Association of Computing Machinery. Vol. 40, No. 3. (1997) 66-72
2. Herlocker, J. L., Konstan, J. A., Terveen, L. G., Riedl, J. T.: Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems. Vol. 22, No. 1. (2004) 5-53
3. Jung, K. Y., Lee, J. H.: User Preference Mining through Hybrid Collaborative Filtering and Content-based Filtering in Recommendation System. IEICE Transaction on Information and Systems. Vol. E87-D, No. 12. (2004) 2781-2790
4. Jung, K. Y., Park, D. H., Lee, J. H.: Personalized Movie Recommender System through Hybrid 2-Way Filtering with Extracted Information. Proc. of the International Conference on Flexible Query Answering Systems. (2004) 473-486
5. Ko, S. J., Lee, J. H.: User Preference Mining through Collaborative Filtering and Content Based Filtering in Recommender System. Proc. of the International Conference on E-Commerce and Web Technologies. (2002) 244-253
6. Kohrs, A., Merialdo, B.: Improving Collaborative Filtering with Multimedia Indexing Techniques to Create User-Adapting Web Sites. Proc. of the ACM International Conference on Multimedia. (1999) 27-36
7. Lee, W. S.: Collaborative Learning for Recommender Systems. Proc. of the 18th International Conference on Machine Learning. (2001) 314-321
8. McJones, P.: EachMovie Dataset: <http://www.research.digital.com/SRC/eachmovie/>.
9. Melville, P., Mooney, R. J., Nagarajan, R.: Content-Boosted Collaborative Filtering for Improved Recommendations. Proc. of the National Conference on Artificial Intelligence. (2002) 187-192
10. Mitchell, T.: *Machine Learning*, McGraw-hill. (1997) 154-200
11. Pazzani, M. J.: A Framework for Collaborative, Content-based and Demographic Filtering. Artificial Intelligence Review. Vol. 13, No. 5-6. (1999) 393-408
12. Wang, J., de Vries, A. P., Reinders, M. J.T.: A User-Item Relevance Model for Log-based Collaborative Filtering. Proc. of the European Conference on Information Retrieval. (2006) 37-48

A Declarative Kernel for $\mathcal{ALC}$ Concept Descriptions

Nicola Fanizzi and Claudia d'Amato

Dipartimento di Informatica, Università degli Studi di Bari
Campus Universitario, Via Orabona 4, 70125 Bari, Italy
{fanizzi, claudia.damato}@di.uniba.it

Abstract. This work investigates on kernels that are applicable to semantic annotations expressed in Description Logics which are the theoretical counterpart of the standard representations for the Semantic Web. Namely, the focus is on the definition of a kernel for the $\mathcal{ALC}$ logic, based both on the syntax and on the semantics of concept descriptions. The kernel is proved to be valid. Furthermore, semantic distance measures are induced from the kernel function.

1 Learning in Multi-relational Settings

Many application domains require structured data representations, spanning from parsing and text categorization of semi-structured text and web-pages to computational biology and drug design. A new emerging application domain is represented by the Semantic Web [1] where knowledge intensive manipulations on complex relational descriptions are foreseen to be performed by machines. In this context, *Description Logics* (DLs) [2] has been adopted as the core technology for ontology languages, such as OWL¹. This family of languages is endowed with well-founded semantics and reasoning services (mostly for deductive inference) descending from a long research line.

Learning in multi-relational settings with traditional logic-based methods is its inherent intractability, unless some language bias is imposed to further constrain the representation. A promising direction is the generation of adequate relational features based on DLs [3]. Yet, for the sake of tractability, often very simple languages are employed. Namely these languages support only existential attributes and conjunction which are well suited for existential descriptions.

Kernel methods [4] are a family of efficient and effective algorithms (including the *support vector machines* – SVMs [5]) that have been applied to a variety of problems, recently also those requiring structured representations [6, 7]. In this paper we propose a kernel function which is suitable for DLs representations. Particularly, we adopt $\mathcal{ALC}$ [2] as a tradeoff between efficiency and expressiveness thus yielding more expressiveness for the relational descriptions that are abstracted by the elicited features. These features, in turn, can be acquired to enrich the starting knowledge base building up a new representation. A mere syntactic approach is not suited for expressive logics. This is the reason why reasoning cannot be performed as merely structural manipulations of the descriptions; rather it needs for tableaux algorithms where semantics comes in [2]. We introduce kernels that are based both on the syntactic structure (by means of the

¹ <http://www.w3.org/TR/owl-ref/>

convolution [8], see the next section) and also on the semantics which can be derived from the knowledge base (in particular, from the ABox). Specifically, we define a kernel function which is applicable to annotations that can be expressed in $\mathcal{ALC}$ and prove it valid and efficiently computable. This makes it eligible for kernel machines. Moreover, since kernels reflect a notion of similarity in an unknown embedded space of features, we can also derive distance measures from them. Kernels and distances will be likely adopted for specific tasks such as classification, clustering and ranking of individuals.

The paper is organized as follows. After recalling the basics of kernels (Sect. 2), we present the DLs representation in Sect. 3. The kernel is presented (Sect. 4) and discussed (Sect. 5). Finally, possible developments of the method are also examined (Sect. 6).

2 Kernel Functions

One of the advantages of kernel methods is that the learning algorithm (inductive bias) and the choice of the kernel (language bias) are almost completely independent. Thus, an efficient algorithm for attribute-value instance spaces can be converted on one suitable for structured spaces (e.g. trees, graphs) by merely replacing the kernel function with a suitable one. This motivates the increasing interest addressed to the SVMs and other kernel methods [4, 5] that reproduce learning in high-dimensional spaces while working like in a vectorial (propositional) representation.

Most algorithms adopt feature representations where an example is represented as a collection of boolean features x with a target label y : $(x, y) \in \{0, 1\}^n \times \{-1, 1\}$. PERCEPTRON is a well-known online algorithm working on such examples. It essentially maintains a weight vector $w \in \mathbb{R}^n$. Per each incoming training example $x \in \{0, 1\}^n$, the algorithm predicts a label according to the linear threshold $w \cdot x \geq 0$. On erroneous predictions, the weights w are revised depending on the examples that provoked the mistake: $w = \sum_{v \in M} l(v)v$, where M is the set of examples on which the algorithms made an error and l is a function returning the label of a given example. The procedure can be summarized by predicting a positive outcome if

$$w \cdot x = \left(\sum_{v \in M} l(v)v \right) \cdot x = \sum_{v \in M} l(v)(v \cdot x) \geq 0$$

Denoting with $\phi(x)$ the transformation of x into an unknown new (highly dimensional) feature space, we can rewrite the relation as: $\sum_{v \in M} l(v)(\phi(v) \cdot \phi(x)) \geq 0$. The *kernel trick* is performed defining a kernel function $k(v, x) = \phi(v) \cdot \phi(x)$, thus obtaining a *Kernel Perceptron* [9]: $\sum_{v \in M} l(v)k(v, x) \geq 0$.

Many different kernels have been proposed for vector spaces. Haussler [8] introduced the general notion of *convolution kernels* to deal with compound data by decomposing structured data into parts, for which valid kernels have already been defined:

$$k_{\text{conv}}(x, y) = \sum_{\substack{\bar{x} \in R^{-1}(x) \\ \bar{y} \in R^{-1}(y)}} \prod_{i=1}^D k_i(\bar{x}_i, \bar{y}_i) \quad (1)$$

where R is a composition relationship building a single compound out of D simpler objects, each from a space that is already endowed with a valid kernel.

Other works have continued this line of research introducing kernels for strings, trees, graphs and other discrete structures [10]. In particular, [6] shows how to define generic kernels based on type construction where types are defined in a declarative way. Furthermore, the above mentioned kernels for vector spaces can be applied to relational kernels in order to act as modifiers.

While these kernels were defined as depending on specific structures, a more flexible method is building kernels as parameterized on a uniform representation. Cumby and Roth [11] propose the syntax-driven definition of kernels based on a simple DLs representation, the *Feature Description Language*. They show that the feature space blow-up is mitigated by the adoption of an efficiently computable kernels. Kernel functions for structured data, parameterized on a description language, allow for the employment of algorithms such as SVM's that can simulate feature generation. These functions transform the initial representation of the instances into the related active features, thus allowing learning the classifier directly from the structured data.

The adopted distance between structures [9] produces classifiers that could be induced by the PERCEPTRON algorithm over the fully blown space of features. Meanwhile, it allows for the definition of a metric over the structures and it enables the control of the feature space size which has an impact on the efficiency of both the learning process and the produced classifier. This is particularly important when the number of examples exceeds the data dimensionality: a case that has been shown disadvantageous for kernel methods [11].

3 Reference Representation Space

In this section, we recall syntax and semantics for the reference representation $\mathcal{ALC}$ [2] adopted in the rest of the paper, since it turns out to be sufficiently expressive to support most of the principal constructors of a markup language for annotations.

In a DLs language, primitive *concepts*, denoted with names taken from $N_C = \{P, P_1, \dots\}$, are interpreted as subsets of a certain domain of objects or equivalently as unary relations on such domain and primitive *roles*, denoted with names taken from $N_R = \{R, S, \dots\}$, are interpreted as binary relations on such a domain (properties). Complex descriptions can be built using primitive concepts and roles by means of the constructors in Table 1. Their semantics is defined by an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is the *domain* of the interpretation and the functor $\cdot^{\mathcal{I}}$ stands for the *interpretation function*, mapping the intension of concepts and roles to their extension.

A *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is made up of a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$. $\mathcal{T}$ is a set of concept definitions $C \equiv D$, meaning $C^{\mathcal{I}} = D^{\mathcal{I}}$, where C is the concept name and D is a description given in terms of the language constructors. $\mathcal{A}$ contains extensional assertions on concepts and roles, e.g. $C(a)$ and $R(a, b)$, meaning, respectively, that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$, i.e., respectively, a is instance of the concept C and b is an R -filler for a . A notion of *subsumption* between concepts is given in terms of the interpretations:

Definition 3.1 (subsumption). *Given two descriptions C and D , C subsumes D , denoted by $C \sqsupseteq D$, iff for every interpretation $\mathcal{I}$ it holds that $C^{\mathcal{I}} \supseteq D^{\mathcal{I}}$.*

Table 1. $\mathcal{ALC}$ constructors and their meaning

NAME	SYNTAX	SEMANTICS
top concept	$\top$	$\Delta^{\mathcal{I}}$
bottom concept	$\perp$	$\emptyset$
concept	C	$C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
concept negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
concept conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
concept disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
existential restriction	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} ((x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$
value restriction	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}} ((x, y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}})\}$

General axioms based on subsumption ($C \sqsubseteq D$) are often also allowed in the T-boxes as partial definitions. Indeed, $C \equiv D$ amounts to $C \sqsubseteq D$ and $D \sqsubseteq C$.

Example 3.1. Some examples of concept definitions in the proposed language are:

Employee $\equiv$ Person $\sqcap \exists \text{worksIn.Company}$

GeneralDirector $\equiv$ Employee $\sqcap \exists \text{worksIn.Company} \sqcap \exists \text{hasPosition.Head}$

which correspond to the sentences: "an employee is a person who works in a company"

and "a general director is an employee who works in a company as a head".

The following are instances of simple assertions:

Person(Lenny), Person(John), GeneralDirector(Tony), worksIn(Lenny, Isoco).

Supposing that Employee $\sqsubseteq$ Person is known (in the TBox), one can infer that: Person(Tony), and then ($\exists \text{worksIn.Company}$)(Tony).

Given these primitive concepts and roles, it is possible to define many other related concepts such as:

Developer $\equiv$ Employee $\sqcap \exists \text{worksIn.Company} \sqcap \exists \text{hasPosition.}\neg \text{Head} \sqcap$

$\sqcap \exists \text{isAppliedTo.}(\text{Project} \sqcap \forall \text{developedIn.ProgLanguage})$ and

JavaDeveloper $\equiv$ Developer $\sqcap \exists \text{isAppliedTo.}(\text{Project} \sqcap \forall \text{developedIn.Java})$

It is easy to see that the following relationships hold:

Employee $\sqsubseteq$ GeneralDirector and Employee $\sqsubseteq$ Developer $\sqsubseteq$ JavaDeveloper. $\square$

One of the most important inference services from the inductive learning viewpoint is *instance checking*, that is deciding whether an individual is an instance of a concept (w.r.t. an ABox). Related to this problem, it is often necessary to solve the *realization problem* that requires to compute, given an ABox and an individual the concepts which the individual belongs to:

Definition 3.2 (most specific concept). *Given an ABox $\mathcal{A}$ and an individual a , the most specific concept of a w.r.t. $\mathcal{A}$ is the concept C , denoted $\text{MSC}_{\mathcal{A}}(a)$, such that $\mathcal{A} \models C(a)$ and $\forall D$ such that $\mathcal{A} \models D(a)$, it holds: $C \sqsubseteq D$.*

In the general case of cyclic A-Boxes represented in an expressive DL endowed with existential restriction the MSC cannot be expressed as a finite description [2], thus it can only be approximated. Since the existence of the MSC for an individual w.r.t. an ABox is not guaranteed, generally an approximation of the MSC is considered up to

a certain depth k , denoted MSC^k . The maximum depth k has been shown to correspond to the depth of the considered ABox. Henceforth, we will indicate generically an approximation to the maximum depth with MSC^* .

Many semantically equivalent (yet syntactically different) descriptions can be given for the same concept. Nevertheless equivalent concepts can be reduced to a normal form by means of rewriting rules that preserve their equivalence, such as: $\forall R.C_1 \sqcap \forall R.C_2 \equiv \forall R.(C_1 \sqcap C_2)$ (see [2] for issues related to normalization and simplification). Some notation is necessary to define the $\mathcal{ALC}$ normal form, naming the different parts of a description: $\text{prim}(C)$ is the set of all the primitive concepts (and their negations) occurring at the top-level of C ; $\text{val}_R(C) = C_1 \sqcap \dots \sqcap C_n$ if there exists a value restriction $\forall R.(C_1 \sqcap \dots \sqcap C_n)$ on the top-level of C , (a single one because of the equivalence mentioned above) otherwise $\text{val}_R(C) = \top$; $\text{ex}_R(C)$ is the set of the descriptions C' appearing in existential restrictions $\exists R.C'$ at the top-level conjunction of C . Now the normal form can formally defined as follows:

Definition 3.3 ($\mathcal{ALC}$ normal form). A description D is in $\mathcal{ALC}$ normal form iff $D \equiv \perp$ or $D \equiv \top$ or if $D = C_1 \sqcup \dots \sqcup C_n$ with

$$C_i = \bigsqcap_{P \in \text{prim}(C_i)} P \sqcap \bigsqcap_{R \in N_R} \left(\forall R.\text{val}_R(C_i) \sqcap \bigsqcap_{E \in \text{ex}_R(C_i)} \exists R.E \right)$$

where, for all $i = 1, \dots, n$, $C_i \not\equiv \perp$ and, for any $R \in N_R$, $\text{val}_R(C_i)$ and every sub-description in $\text{ex}_R(C_i)$ are in normal form.

Given an $\mathcal{ALC}$ description in normal form, a related $\mathcal{ALC}$ description tree is an AND-OR tree. It alternates disjunctive ($\sqcup$) and conjunctive ($\sqcap$) nodes and it is rooted in a disjunctive node (according to the normal form). Conjunctive nodes are labeled by a $\text{prim}(\cdot)$ set and, per each $R \in N_R$, branching one edge labeled $\forall R$ to the subtree of $\text{val}_R(\cdot)$ and for each $E \in \text{ex}_R(\cdot)$, an edge labeled $\exists R$ to the subtree related to E . Note that an empty label in a node is equivalent to the top concept ($\top$).

4 A Relational Kernel for $\mathcal{ALC}$

We elaborate on [6] in order to obtain a family of valid kernels for the space X of $\mathcal{ALC}$ descriptions. Then we will induce a distance suitable for further uses.

One possibility for defining a kernel for $\mathcal{ALC}$ descriptions is based on the AND-OR tree structure of the descriptions, like for the standard tree kernels [6] where similarity between trees depends on the number of similar subtrees (or paths unraveled from such trees). Yet this would end in a merely syntactic measure which does not fully capture the semantic nature of expressive DLs languages such as $\mathcal{ALC}$. Recurring to the idea of the convolution kernels [8], we use the normal form to decompose complex descriptions level-wise into sub-descriptions. There are three possibilities for each level: the level is dominated by the disjunction of concepts, it is a conjunction of concepts or it is simply a level of primitive concepts.

Definition 4.1 ($\mathcal{ALC}$ kernel). *Given two descriptions in normal form $D_1 = \bigsqcup_{i=1}^n C_i^1$ and $D_2 = \bigsqcup_{j=1}^m C_j^2$, and an interpretation $\mathcal{I}$, the $\mathcal{ALC}$ kernel based on $\mathcal{I}$ is the function $k_{\mathcal{I}} : X \times X \mapsto \mathbb{R}$ inductively defined as follows.*

disjunctive descriptions: $k_{\mathcal{I}}(D_1, D_2) = \lambda \sum_{i=1}^n \sum_{j=1}^m k_{\mathcal{I}}(C_i^1, C_j^2)$ with $\lambda \in]0, 1]$

conjunctive descriptions:

$$k_{\mathcal{I}}(C^1, C^2) = \prod_{\substack{P_1 \in \text{prim}(C^1) \\ P_2 \in \text{prim}(C^2)}} k_{\mathcal{I}}(P_1, P_2) \cdot \prod_{R \in N_R} k_{\mathcal{I}}(\text{val}_R(C^1), \text{val}_R(C^2)) \cdot \prod_{R \in N_R} \sum_{\substack{C_i^1 \in \text{ex}_R(C^1) \\ C_j^2 \in \text{ex}_R(C^2)}} k_{\mathcal{I}}(C_i^1, C_j^2)$$

primitive concepts: $k_{\mathcal{I}}(P_1, P_2) = k_{\text{set}}(P_1^{\mathcal{I}}, P_2^{\mathcal{I}}) = |P_1^{\mathcal{I}} \cap P_2^{\mathcal{I}}|$

where k_{set} is the kernel for set structures defined in [6]. This case includes also the negation of primitive concepts using: $(\neg P)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus P^{\mathcal{I}}$

The rationale for this kernel is that similarity between disjunctive descriptions is treated by taking the sum of the cross-similarities between any couple of disjuncts from either description. The term λ is employed to downweight the similarity of the sub-descriptions on the grounds of the level where they occur. As regards the conjunctive kernel it computes the similarity between two input descriptions, distinguishing among primitive concepts, those referred in the value restrictions and those referred in the existential restrictions. These similarity values are multiplied reflecting the fact that all the restrictions have to be satisfied at a conjunctive level. The similarity between primitive concepts is measured in terms of the intersection of their extension.

Remind that the *unique names assumption* is made on the individuals occurring in the ABox $\mathcal{A}$. Hence, we can take into account the canonical interpretation $\mathcal{I}$, using $\text{Ind}(\mathcal{A})$ as its domain ($\Delta^{\mathcal{I}} := \text{Ind}(\mathcal{A})$). Therefore, the kernel can be specialized as follows: since the kernel for primitive concepts is essentially a set kernel we can set the constant λ_p to $1/\Delta^{\mathcal{I}}$ so that the cardinality of the intersection is weighted by the number of individuals occurring in the overall ABox. Alternatively, another choice could be $\lambda_P = 1/|P_1^{\mathcal{I}} \cup P_2^{\mathcal{I}}|$ which would weight the rate of similarity (the extension intersection) measured by the kernel with the size of the concepts measured in terms of the individuals belonging to their extensions.

Example 4.1. Consider the following descriptions (whose trees are depicted in Fig. 1): $C \equiv (P_1 \sqcap P_2) \sqcup (\exists R.P_3 \sqcap \forall R.(P_1 \sqcap \neg P_2))$ and $D \equiv P_3 \sqcup (\exists R.\forall R.P_2 \sqcap \exists R.\neg P_1)$ we employed the parentheses to emphasize the level-wise structure of the descriptions. Now, suppose $P_1^{\mathcal{I}} = \{a, b, c\}$, $P_2^{\mathcal{I}} = \{b, c\}$, $P_3^{\mathcal{I}} = \{a, b, d\}$ and $\Delta^{\mathcal{I}} = \{a, b, c, d, e\}$. The top-level is normally disjunctive so we have to compute: $k_{\mathcal{I}}(C, D) = \lambda \sum_{i=1}^2 \sum_{j=1}^2 k_{\mathcal{I}}(C_i, D_j)$ where $C_1 \equiv P_1 \sqcap P_2$, $C_2 \equiv \exists R.P_3 \sqcap \forall R.(P_1 \sqcap \neg P_2)$, $D_1 \equiv P_3$ and $D_2 \equiv \exists R.\forall R.P_2 \sqcap \exists R.\neg P_1$.

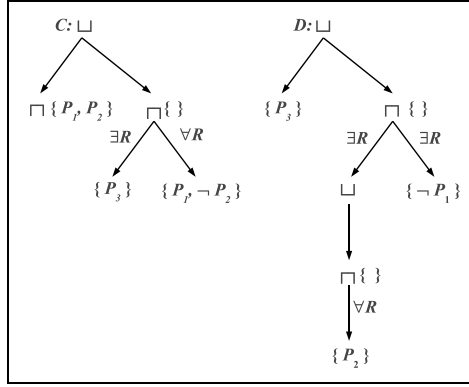


Fig. 1. The (compacted) tree representation for the descriptions used in Example 4.1

Now, let us compute: $k_{\mathcal{I}}(C_1, D_1) = \prod_{P_1^C \in \text{prim}(C_1)} \prod_{P_1^D \in \text{prim}(D_1)} k_{\mathcal{I}}(P_1^C, P_1^D) = k_{\mathcal{I}}(P_1, P_3) \cdot k_{\mathcal{I}}(P_2, P_3) = |\{a, b, c\} \cap \{a, b, d\}| \cdot |\{b, c\} \cap \{a, b, d\}| = 2 \cdot 1 = 2$

No contribution comes from the value and existential restrictions: the factors amount to 1 since $\text{val}_R(C_1) = \text{val}_R(D_1) = \top$ and $\text{ex}_R(C_1) = \text{ex}_R(D_1) = \emptyset$ which make that equivalent to $\top$ too.

Now, we compute:

$$\begin{aligned} k_{\mathcal{I}}(C_1, D_2) &= k_{\mathcal{I}}(P_1, \top) \cdot k_{\mathcal{I}}(P_2, \top) \cdot \prod_{R \in N_R} k_{\mathcal{I}}(\text{val}_R(C_1), \text{val}_R(D_2)) \cdot \\ &\prod_{R \in N_R} \sum_{E_C \in \text{ex}_R(C_1)} \sum_{E_D \in \text{ex}_R(D_2)} k_{\mathcal{I}}(E_C, E_D) = (3 \cdot 2) \cdot k_{\mathcal{I}}(\top, \top) \cdot (k_{\mathcal{I}}(\top, \forall R.P_2) + \\ &k_{\mathcal{I}}(\top, \neg P_1)) = 6 \cdot 5 \cdot (\lambda \sum_{C' \in \{\top\}} \sum_{D' \in \{\forall R.P_2\}} k_{\mathcal{I}}(C', D') + 2) = 30 \cdot (\lambda \cdot 1 \cdot k_{\mathcal{I}}(\top, P_2) \cdot \\ &1 + 2) = 30 \cdot (2\lambda + 2) = 60(\lambda + 1). \end{aligned}$$

Note that empty prim is equivalent to $\top$ and in one case we had to recur one more level in depth.

$$\begin{aligned} \text{Now, we compute: } k_{\mathcal{I}}(C_2, D_1) &= k_{\mathcal{I}}(\top, P_3) \cdot \prod_{R \in N_R} k_{\mathcal{I}}(\text{val}_R(C_2), \top) \cdot \\ &\prod_{R \in N_R} \sum_{E_C \in \text{ex}_R(C_2)} \sum_{E_D \in \text{ex}_R(D_1)} k_{\mathcal{I}}(E_C, E_D) = 3 \cdot k_{\mathcal{I}}(P_1 \sqcap \neg P_2, \top) \cdot k_{\mathcal{I}}(P_3, \top) = \\ &3 \cdot \lambda (k_{\mathcal{I}}(P_1, \top) \cdot k_{\mathcal{I}}(\neg P_2, \top)) \cdot 3 = 3 \cdot \lambda (3 \cdot 3) \cdot 3 = 81\lambda. \end{aligned}$$

Note that the absence of prim set is equivalent to $\top$ and since one of the sub-concepts has no existential restriction the product gives no contribution. Finally, we compute:

$$\begin{aligned} k_{\mathcal{I}}(C_2, D_2) &= k_{\mathcal{I}}(\top, \top) \cdot \prod_{R \in N_R} k_{\mathcal{I}}(P_1 \sqcap \neg P_2, \top) \cdot \\ &\prod_{R \in N_R} \sum_{C'' \in \{P_3\}} \sum_{D'' \in \{\forall R.P_2, \neg P_1\}} k_{\mathcal{I}}(C'', D'') = 5 \cdot \lambda (k_{\mathcal{I}}(P_1, \top) \cdot k_{\mathcal{I}}(\neg P_2, \top)) \cdot \\ &(k_{\mathcal{I}}(P_3, \forall R.P_2) + k_{\mathcal{I}}(P_3, \neg P_1)) = 5\lambda(3 \cdot 3) \cdot (\lambda k_{\mathcal{I}}(P_3, \top) \cdot k_{\mathcal{I}}(\top, P_2) \cdot k_{\mathcal{I}}(\top, \top) + 1) = \\ &45\lambda(\lambda \cdot 3 \cdot 2 \cdot 5 + 1) = 45\lambda(30\lambda + 1) \end{aligned}$$

Again the absence of restrictions was evaluated as the top concept.

$$\text{By collecting the four intermediate results we have: } k_{\mathcal{I}}(C, D) = 2 + 60(\lambda + 1) + 81\lambda + 45\lambda(30\lambda + 1) = 2 + 60\lambda + 60 + 81\lambda + 1350\lambda^2 + 45\lambda = 62 + 186\lambda + 1350\lambda^2 \quad \square$$

Observe that the function could be also specialized to take into account the similarity between different relationships. This would amount to considering each couple of existential and value restriction with one element from each description (or equivalently from each related AND-OR tree) and the computing the convolution of the

sub-descriptions in the restriction. As previous suggested for λ , this should be weighted by a measure of similarity between the roles measured on the grounds of the available semantics. We propose therefore the following weight: given two roles $R, S \in N_R$: $\lambda_{RS} = |R^{\mathcal{I}} \cap S^{\mathcal{I}}|/|\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}|$

As suggested before, the intersection could be measured on the grounds of the relative role extensions with respect to the whole domain of individuals, as follows: $\lambda_{RS} = |R^{\mathcal{I}} \cap S^{\mathcal{I}}|/|R^{\mathcal{I}} \cup S^{\mathcal{I}}|$. It is also worthwhile to recall that some DLs knowledge bases support also the so called *R-box* [2] with assertions concerning the roles, thus we might know beforehand that for instance $R \sqsubseteq S$ and compute their similarity consequently.

The kernel can be extended to the case of individuals $a, b \in \text{Ind}(\mathcal{A})$ simply by taking into account the approximations of their most specific concepts:

$$k_{\mathcal{I}}(a, b) = k_{\mathcal{I}}(\text{MSC}^*(a), \text{MSC}^*(b))$$

In this way, we move from a graph representation like the ABox portion containing an individual to an intensional tree-structured representation (investigated in this paper).

5 Discussion

The validity of a kernel depends on the fact that the function is *positive definite*:

for any $n \in \mathbb{Z}^+$, $x_1, \dots, x_n \in X$ and $(c_1, \dots, c_n) \in \mathbb{R}^n$: $\sum_{i,j=1}^n c_i c_j k(x_i, x_j) \geq 0$. Positive definiteness can be also proven exploiting some closure properties of the class of positive definite kernel functions [8]. Namely, multiplying a kernel by a constant, adding or multiplying two kernels yields another valid kernel. We can demonstrate that the function introduced above is indeed a valid kernel for our space of hypotheses. Observe that the core function is the one on primitive concept extensions. It is essentially a set kernel [6]. The versions for top-level conjunctive and disjunctive descriptions are also positive definite being essentially based on the primitive kernel. Descending through the levels there is an interleaving of the employment of these function up to the basic case of the function for primitive descriptions.

Proposition 5.1. *The function $k_{\mathcal{I}}$ is a valid kernel for the space X of $\mathcal{ALC}$ descriptions.*

Proof. The validity is verified by proving symmetry and positive definiteness for $k_{\mathcal{I}}$.

(symmetry) The kernel function $k_{\mathcal{I}}$ is symmetric because of the commutativity of the operations involved in its computation of the three functions.

(positive definiteness) This can be proven by structural induction on the depth d of the descriptions (maximum number of levels).

- ($d = 0$) This case is related to the function for primitive descriptions (represented as a flat tree). It is a convolution based on the set kernel (see [6]) between single primitive concept extensions. The multiplication by a constant λ does not affect this property.
- ($d > 0$) Assuming by hypothesis that $k_{\mathcal{I}}$ is positive definite for descriptions of depth up to $d - 1$. Now, if the descriptions are in normal form $k_{\mathcal{I}}$ can be computed based on the disjunctive or the conjunctive version depending on the top-level.

In the former case, we have a convolution [8] of a function for simpler descriptions (maximum depth $< d$) which is positive definite by hypothesis of induction. Hence it must be positive definite for this case also, by the closure of the class of positive definite functions.

In the latter case, we have a product of positive definite functions for primitive (first factor) or sums of positive definite functions for simpler descriptions (maximum depth $< d$) which is positive definite by hypothesis of induction. Again, by the closure of the class of positive definite functions, the function

By induction, then, the function is positive definite for descriptions of any depth $\square$

As regards the efficiency, it is possible to show that the kernel function can be computed in time $O(|N_1||N_2|)$ where $|N_i|$, $i = 1, 2$, is the number of nodes of the concept AND-OR trees. It can be computed by means of dynamic programming.

Given a kernel function, it is very straightforward to define also an induced distance measure [6] that, differently from other proposed measures, is based also on the underlying semantics provided by the ABox assertions:

Definition 5.1 (induced distance). *Given two descriptions C and D , and the canonical interpretation $\mathcal{I}_A$, the induced distance based on the $\mathcal{ALC}$ kernel $k_{\mathcal{I}}$, is the function $d_{\mathcal{I}} : X \times X \mapsto \mathbb{R}$ defined as follows: $d_{\mathcal{I}}(C, D) = \sqrt{k_{\mathcal{I}}(C, C) - 2k_{\mathcal{I}}(C, D) + k_{\mathcal{I}}(D, D)}$.*

Being $k_{\mathcal{I}}$ a valid kernel, it follows that the induced distance $d_{\mathcal{I}}$ is a metric [6]: (1) $d_{\mathcal{I}}(C, D) \leq d_{\mathcal{I}}(C, E) + d_{\mathcal{I}}(E, D)$, (2) $d_{\mathcal{I}}(C, D) = d_{\mathcal{I}}(D, C)$, (3) $C = D \Leftrightarrow d_{\mathcal{I}}(C, D) = 0$.

As proposed in [8], the normalization of kernels to the range $[0, 1]$ can be very important in practical cases. By normalizing our kernel function as follows:

$$\tilde{k}_{\mathcal{I}}(C, D) = \frac{k_{\mathcal{I}}(C, D)}{\sqrt{k_{\mathcal{I}}(C, C)}\sqrt{k_{\mathcal{I}}(D, D)}}$$

we obtain a valid kernel such that: $0 \leq \tilde{k}_{\mathcal{I}}(C, D) \leq 1$ hence another distance can be induced in the following way:

$$d_{\mathcal{I}}^2(C, D) = \frac{1}{2}(\log \tilde{k}_{\mathcal{I}}(C, C) + \log \tilde{k}_{\mathcal{I}}(D, D)) - \log \tilde{k}_{\mathcal{I}}(C, D)$$

which is the *radial distance* induced from the kernel.

As mentioned before, these distances may have lots of practical applications spanning from instance-based classification and clustering to raking of query answers retrieved in a knowledge bases. This constitutes a new approach in the Semantic Web area. Indeed in this way it is possible to combine the efficiency of the numerical approaches and the effectiveness of a symbolic representation. Instead currently, almost all tasks applied to the Semantic Web area use a logic-based approach, mainly based on deductive reasoning. Moreover the presented function can be used in inductive tasks i.e. classify individuals of an A-Box. This can help to make complete large ontology that are incomplete hence it is possible to apply deductive reasoning to such complete knowledge bases.

The main well of complexity for the presented function is the computation of the concepts extensions. However this complexity could be easily decreased by pre-calculating

the extension of all primitive concepts, hence using these as sets and so apply set operations.

6 Conclusions and Future Work

We presented kernels for a DLs language which is sufficiently expressive to stand as the theoretical counterpart for the standard representations for annotations in the Semantic Web. We focussed on the definition of a kernel for *ALC* descriptions, that, differently from other kernel functions for structured data in the literature, are both syntax and semantic-driven. This kernel function is proved to be a valid, thus representing the basis for the definition of semantic distances suitable for the same representation. The distances could be employed in unsupervised as well as supervised methods for learning from annotations. Particularly a classifier of individuals asserted in an ontology is under development and the first experimental results are proving promising the proposed approach. The main weakness of the approach is on its scalability towards more complex DL languages. While computing MSC approximations might be feasible, it may be more difficult to define a normal form for comparing descriptions. Indeed, as long as the expressive power increases, it also becomes more redundant, hence the semantics becomes more and more detached from the syntactic structure of the descriptions. This work could be extended towards more expressive languages for a full support to the OWL ontology language. For instance the target language could be enriched with (qualified) number restrictions and inverse roles, like in *SHIQ* [2] which is the language in the DLs family that includes all the constructors employed in OWL.

References

- [1] Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* **284** (2001) 34–43
- [2] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: *The Description Logic Handbook*. Cambridge University Press (2003)
- [3] Cumby, C.M., Roth, D.: Learning with feature description logics. In Matwin, S., Sammut, C., eds.: *Proceedings of the 12th International Conference on Inductive Logic Programming, ILP2002*. Volume 2583 of LNCS., Springer (2003) 32–47
- [4] Schölkopf, B., Smola, A.: *Learning with Kernels*. The MIT Press (2002)
- [5] Cristianini, N., Shawe-Taylor, J.: *An Introduction to Support Vector Machines*. Cambridge University Press (2000)
- [6] Gärtner, T., Lloyd, J., Flach, P.: Kernels and distances for structured data. *Machine Learning* **57** (2004) 205–232
- [7] Passerini, A., Frasconi, P., Raedt, L.D.: Kernels on prolog proof trees: Statistical learning in the ILP setting. *Journal of Machine Learning Research* **7** (2006) 307–342
- [8] Haussler, D.: Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, Department of Computer Science, University of California – Santa Cruz (1999)
- [9] Khardon, R., Roth, D., Servedio, R.: In: *Efficiency versus convergence of boolean kernels for on-line learning algorithms*. The MIT Press (2002)
- [10] Gärtner, T.: A survey of kernels for structured data. *SIGKDD Explorations* **5** (2003) 49–58
- [11] Cumby, C., Roth, D.: On kernel methods for relational learning. In Fawcett, T., N. Mishra, eds.: *Proceedings of the 20th International Conference on Machine Learning, ICML2003*, AAAI Press (2003) 107–114

RDF as Graph-Based, Diagrammatic Logic

Frithjof Dau

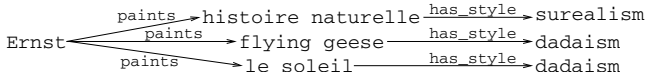
Dept. of Mathematics, Dresden Technical University, Germany

Abstract. The Resource Description Framework (RDF) is the basic standard for representing information in the Semantic Web. It is mainly designed to be machine-readable and -processable. This paper takes the opposite side of view: RDF is investigated as a logic system designed for the needs of humans. RDF is developed as a logic system based on mathematical graphs, i.e., as *diagrammatic reasoning system*. As such, is has humanly-readable, diagrammatic representations. Moreover, a sound and complete calculus is provided. Its rules are suited to act on the diagrammatic representations. Finally, some normalforms for the graphs are introduced, and the calculus is modified to suit them.

1 Introduction: RDF and Graphs

The Resource Description Framework (RDF) is the basic standard for representing information in the Semantic Web. The information is encoded in statements, which are subject-predicate-object triples of information. RDF can be seen from two perspectives: computers and humans. For computers, it is important that RDF is a machine processable data-format. Thus, much effort is spent on specifications which suit this purpose (e.g. RDF/XML). An approach to make RDF accessible for humans can already be found in the RDF-primer [8], where the set of triples is converted into a mathematical graph: The subjects and objects of all statements are mapped to vertices, and each statement is mapped to a directed edge between its subject and object, labeled by its predicate. Consider the following set of triples $\mathcal{T}_1$ and its representation as graph:

(Ernst,paints,hist. naturelle),(Ernst,paints,flying geese),
(Ernst,paints,le soleil),(hist. naturelle,has style,surrealism),
(flying geese,has style,dadaism),(le soleil,has style,dadaism)



The translation of triple set to these graphs is appealing, but it has two problems.

RDF allows a statement predicate to appear as the subject of another statement. E.g., $\{(\text{type type prop}), (\text{subject type prop})\}$ is a well-formed set of triples, but it cannot be transformed to a graph with the conventions of [8]. The first goal of the paper is to close the gap that not all triple sets can be represented as graphs. For this, we adopt the approach of [1, 2] instead: RDF triple sets are translated to hyper graphs, where subjects, objects, and predicates are

mapped to vertices, and each statement (s, p, o) is mapped to an 3 -ary *hyperedge* linking these vertices.

RDF is used for *information-processing*. In [3] an entailment relation between triple sets is introduced. But this relation does not the diagrammatic representation of RDF. It is well accepted that diagrams are often easier to comprehend than symbolic notations ([7, 6]). The second goal of the paper is, after adopting the semantics of [3], to provide an adequate *diagrammatic* calculus.

In this understanding, RDF is developed as diagrammatic reasoning system (DRS), in a humanly readable and mathematically precise manner. In the next sections, an overview of the syntax, semantics, a sound and complete calculus, and some normalforms of RDF Graphs for RDF as DRS is provided. Due to space limitations, some formal definitions and all proofs are omitted. They can be found in a technical report on www.dr-dau.net.

2 Syntax

In this section we define the syntax of the system, starting with the vocabulary.

Definition 1. Let $\text{Blanks} := \{\_1, \_2, \_3, \dots\}$ be a set of so-called **BLANKS**. A triple $\text{Voc} := (\text{URIs}, \text{TypedLit}, \text{PlainLit})$ is called **VOCABULARY**. The elements of these sets are called **UNIVERSAL RESOURCE IDENTIFIERS**, **TYPED LITERALS**, and **PLAIN LITERALS**, resp. We set $\text{Lit} := \text{TypedLit} \cup \text{PlainLit}$. The elements of Lit are called **LITERALS**. The elements of $\text{URIs} \cup \text{TypedLit} \cup \text{PlainLit}$ are called **NAMES**. We assume that **Blanks**, **URIs**, **TypedLit**, **PlainLit** are pairwise disjoint.

Next we define the well-known triple notation of RDF as well as the corresponding mathematical graphs. To avoid confusion, we use the term *RDF triple set* instead of the common term *RDF graph* for the triple notation.

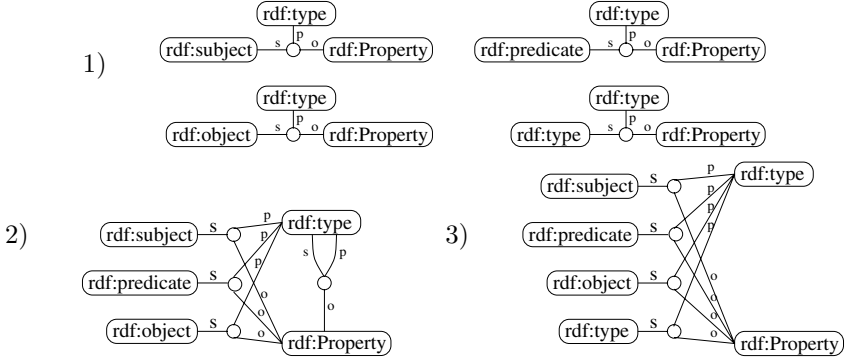
Definition 2. An **RDF TRIPLE SET OVER Voc** is a set of triples (s, p, o) with $s \in \text{URIs} \cup \text{Blanks}$, $p \in \text{URIs}$, $o \in \text{URIs} \cup \text{Blanks} \cup \text{Lit}$. We call s the **SUBJECT**, p the **PROPERTY** and o the **OBJECT** of the triple.

A structure (V, E, ν, κ) is an **RDF GRAPH OVER Voc** iff V and E are finite sets of **VERTICES** and **EDGES**, $\nu : E \rightarrow V^3$ is a mapping such that each vertex is incident with at least one edge, and $\kappa : V \rightarrow \text{Voc}$ is a mapping such that for each $e \in E$, we have $\kappa(e|_1) \in (\text{URIs} \cup \text{Blanks})$, $\kappa(e|_2) \in \text{URIs}$, and $\kappa(e|_3) \in (\text{URIs} \cup \text{Blanks} \cup \text{Lit})$. For $e \in E$ with $\nu(e) = (v_1, v_2, v_3)$ we will often write $e = (v_1, v_2, v_3)$, and each v_i , $i = 1, 2, 3$ is denoted by $e|_i$. For $v \in V$ we set $E_v := \{e \in E \mid \exists i. \nu(e)|_i = v\}$, and for $e \in E$ we set $V_e := \{v \in V \mid \exists i. \nu(e)|_i = v\}$. The set $\{bl \in \text{Blanks} \mid \exists v \in V : \kappa_b(v) = bl\}$ is called the **SET OF BLANKS OF $\mathcal{G}$** .

Compared to RDF triple sets, RDF graphs provide a richer means to express in different ways some given amount of information, and predicates which appear as the subject of other statements do not cause problems. This shall be exemplified with the first four axiomatic triples from the RDF-semantics (see [3]). They are:

```
(rdf:type rdf:type rdf:Property)      (rdf:subject rdf:type rdf:Property)
(rdf:predicate rdf:type rdf:Property) (rdf:object rdf:type rdf:Property)
```


Note that **type** occurs both as subject and as predicate in these triples, so these four triples cannot be displayed due to the graph-drawing conventions of [5]. But with the herein defined RDF graphs, there are now different possibilities to transform this set of triples to a graph. Three of them are depicted below.



In the first graph, each vertex is incident exactly once with an edge: It has so-to-speak a maximum number of vertices. Graphs which satisfy this conditions will be said to be in **ANTI-NORMAL-FORM (ANF)**. For the third graph, we have an opposite situation: No different vertices are labeled the same, i.e., this graph contains the minimal number of vertices. Such graphs are said to be in **NORMAL-FORM (NF)**. For graphs in ANF or NF, we assume moreover that they do not have **REDUNDANT EDGES**, i.e., for all edges $e = (v_1, v_2, v_3)$ and $f = (w_1, w_2, w_3)$ with $\kappa(v_i) = \kappa(w_i)$ for $i = 1, 2, 3$, we have $e = f$.

Let $\mathcal{T}$ be an RDF triple set over Voc . Then let $\Phi_N(\mathcal{T}) := (V, E, \nu, \kappa)$ be the following RDF graph in NF: For each name or blank x in $\mathcal{T}$, let v_x be fresh¹ vertex. Let V be the set of these vertices. Let $E := \mathcal{T}$. For $e = (s, p, o) \in E$, let $\nu(e) = (v_s, v_p, v_o)$. For $v = v_x \in V$, let $\kappa(v_x) := x$. Similarly, let $\Phi_{AN}(\mathcal{T}) := (V, E, \nu, \kappa)$ be the following RDF graph in ANF: For each edge $e = (s, p, o) \in E$, let e_s, e_p, e_o be fresh vertices. Let V be the set of these vertices, let $E := \mathcal{T}$, and for $e = (s, p, o) \in E$, let $\nu(e) := (e_s, e_p, e_o)$. For $v = e_x \in V$, we set $\kappa(v_x) := x$.

Now let $\mathcal{G} := (V, E, \nu, \kappa)$ be an RDF Graph. Let $\Psi(\mathcal{G})$ be the following RDF triple set: $\Psi(\mathcal{G}) := \{(\kappa(e|_1), \kappa(e|_2), \kappa(e|_3)) \mid e \in E\}$.

Each RDF graph in NF resp. in ANF is already been sufficiently described by the set of triples $(\kappa(e|_1), \kappa(e|_2), \kappa(e|_3))$ with $e \in E$. If $\mathcal{T}$ is an RDF triple set, we have $\mathcal{T} = \Psi(\Phi_N(\mathcal{T}))$ and $\mathcal{T} = \Psi(\Phi_{AN}(\mathcal{T}))$. Vice versa: If $\mathcal{G}$ is an RDF graph in NF, we have $\mathcal{G} = \Phi_N(\Psi(\mathcal{G}))$, and if $\mathcal{G}$ is an RDF graph in ANF, we have $\mathcal{G} = \Phi_{AN}(\Psi(\mathcal{G}))$. So the mappings Φ_N , Φ_{AN} and Ψ can be understood as translations between graphs in NF or ANF and triple sets.

A **SUBGRAPH** of an RDF triple set is simply a subset. A **SUBGRAPH** of an RDF graph (V, E, ν, κ) is an RDF Graph (V', E', ν', κ') with $V' \subseteq V$, $E' \subseteq E$, $\nu' = \nu|_{E'}$ and $\kappa' = \kappa|_{V'}$. If we have moreover $E_v \subseteq E'$ for each $v \in V'$, the subgraph is called **CLOSED**.

¹ A vertex or edge x is called **FRESH** iff it is not already an element of the set of vertices and edges we consider.

We will implicitly identify RDF Graphs if they differ only in the names of the occurring blanks. As in [3], graphs like these are called EQUIVALENT.

Finally, we have to define the JOIN and MERGE of RDF graphs. The join of RDF triple sets is defined in [3] to be the set-theoretical union. But only if the joined RDF triple sets have no blanks in common, their join corresponds to their logical conjunction. Then one speaks of the MERGE of the RDF triple sets. Analogously, if $\mathcal{G}_i := (V_i, E_i, \nu_i, \kappa_i)$ is for $i = 1, \dots, n$ an RDF graph such that that all V_i and E_i , $i = 1, \dots, n$, are pairwise disjoint, the JOIN OF THE GRAPHS $\mathcal{G}_i$ is defined to be the RDF graph $\mathcal{G} := (V, E, \nu, \kappa)$ with $V := \bigcup_i V_i$, $E := \bigcup_i E_i$, $\nu := \bigcup_i \nu_i$ and $\kappa := \bigcup_i \kappa_i$. If the set of blanks of the graphs $\mathcal{G}_i$ are pairwise disjoint, the join of the graphs $\mathcal{G}_i$ is also called MERGE of the graphs $\mathcal{G}_i$.

3 Semantics

In this section, the semantics for RDF graphs is defined. It is based on mathematical model theory and adopted from [3, 4] for RDF triple sets.

Definition 3. A MODEL $\mathcal{I}$ FOR $\text{Voc} := (\text{URIs}, \text{TypedLit}, \text{PlainLit})$ is a structure $(\text{IR}, \text{IP}, \text{IEXT}, \text{IS}, \text{IL})$ where IR is a set of RESOURCES, called the DOMAIN or UNIVERSE of $\mathcal{I}$, IP is a set of PROPERTIES of $\mathcal{I}$, $\text{IEXT} : \text{IP} \rightarrow \mathfrak{P}(\text{IR} \times \text{IR})$ is a mapping, $\text{IS} : \text{URIs} \rightarrow \text{IR} \cup \text{IP}$ is a mapping, and $\text{IL} : \text{Lit} \rightarrow \text{IR}$ is a mapping.

Note that the relationship between IR and IP is not specified. From the viewpoint of mathematical logic, one would assume that IR and IP are disjoint. From the RDF viewpoint, it is quite normal to assume that $\text{IP} \subseteq \text{IR}$ holds. Both cases and arbitrary other relations between IR and IP are allowed.

In the next definition, RDF graphs are evaluated in models. We do not assume that the graph and model are based on the same vocabulary. But if a name in a graph is not interpreted in the model, the semantics will always yield that the graph evaluates to false. Thus we will usually assume that the vocabularies of a graph and an interpretation are the same.

Definition 4. Let $\mathcal{G} := (V, E, \nu, \kappa)$ be an RDF graph over $\text{Voc}_{\mathcal{G}}$ and let $\mathcal{I} := (\text{IR}, \text{IP}, \text{IEXT}, \text{IS}, \text{IL})$ be an interpretation over $\text{Voc}_{\mathcal{I}}$. A function $I : V \rightarrow \text{IR}$ is an INTERPRETATION FUNCTION (FOR $\mathcal{G}$ IN $\mathcal{I}$), iff for all $v_1, v_2 \in V$ with $\kappa(v_1) = \kappa(v_2)$, we have $I(v_1) = I(v_2)$, and for each each name $n \in \text{URIs}_{\mathcal{I}} \cup \text{PlainLit}_{\mathcal{I}} \cup \text{TypedLit}_{\mathcal{I}}$ and each vertex $v \in V$ with $\kappa(v) = n$, we have $I(v) = (\text{IL} \cup \text{IR})(n)$.

We say the GRAPH $\mathcal{G}$ HOLDS IN THE MODEL $\mathcal{I}$ and write $\mathcal{I} \models \mathcal{G}$, iff for each $v \in V$, $\kappa(v)$ is a name of $\text{Voc}_{\mathcal{I}}$ or a blank, and there exists an interpretation function $I : V \rightarrow \text{IR}$ such that for each edge $e = (v_1, v_2, v_3) \in E$ and $s := \kappa(v_1)$, $p := \kappa(v_2)$, $o := \kappa(v_3)$, we have $(I(s), I(o)) \in \text{IEXT}(I(p))$ and $I(p) \in \text{IP}$.

If $\mathcal{G}_a, \mathcal{G}_b$ are RDF graphs such that $\mathcal{I} \models \mathcal{G}_b$ holds for each $\mathcal{I}$ with $\mathcal{I} \models \mathcal{G}_a$, we say that $\mathcal{G}_a$ (SEMANTICALLY) ENTAILS $\mathcal{G}_b$, and write $\mathcal{G}_a \models \mathcal{G}_b$.

4 Calculus

We provide a calculus with five rules (which should be understood to manipulate the diagrams of RDF graphs) for Gs. Let $\mathcal{G} := (V, E, \nu, \kappa)$ be given.

Erasing an edge: Let e be an edge. Then e may be erased (and each vertex v which is only incident with e has to be erased as well). That is, we construct the graph $\mathcal{G}^e := (V^e, E^e, \nu^e, \kappa^e)$ with $V^{(e)} := V \setminus \{v \in V_e \mid v \notin V_f \text{ for any } f \in E \text{ with } f \neq e\}$, $E^{(e)} := E \setminus \{e\}$, $\nu^{(e)} := \nu|_{E^e}$, and $\kappa^{(e)} := \kappa|_{E^e}$.

Generalizing a label: Let $bl \in \text{Blanks}$ be a blank which does not appear in $\mathcal{G}$. Let $V' \subseteq V$ be a set of vertices which are identically labeled, i.e., we have $\kappa(v_1) = \kappa(v_2)$ for all $v_1, v_2 \in V'$. Then, for each $v \in V$, $\kappa(v)$ may be replaced by $\kappa(v) := bl$. That is, we construct the graph $\mathcal{G}^g := (V, E, \nu, \kappa^g)$ with $\kappa^g(w) := \kappa(w)$ for all $w \notin V'$ and $\kappa^g(v) := bl$ for all $w \in V'$.

Merging two vertices: Let $v_1 \neq v_2$ be two vertices with $\kappa(v_1) = \kappa(v_2)$. Then v_2 may be merged into v_1 (i.e., v_2 is erased and, for every edge $e \in E$, $e|_i = v_2$ is replaced by $e|_i = v_1$). That is, we construct the graph $\mathcal{G}^m := (V^m, E, \nu^m, \kappa)$ with $V^m := V \setminus \{v_2\}$, and for all $e \in E$ and $i = 1, 2, 3$ we have $\nu^m(e)(i) := \nu(e)(i)$, if $\nu(e)(i) \neq v_2$, and $\nu^m(e)(i) := v_1$ else.

Splitting a vertex: Let v_1 be a vertex, incident with edges $e_1, \dots, e_n$. Then we can insert a fresh vertex v_2 with $\kappa(v_2) := \kappa(v_1)$, and on $e_1, \dots, e_n$, arbitrary occurrences of v_1 may be replaced by v_2 . That is, we construct a graph $\mathcal{G}^i := (V^i, E, \nu^i, \kappa)$ with $V^m := V \cup \{v_2\}$, for all $e \in E$ and $i = 1, 2, 3$ we have $\nu^m(e)(i) := \nu(e)(i)$, if $\nu(e)(i) \neq v_1$ and $\nu^m(e)(i) \in \{v_1, v_2\}$ else, and which satisfies $E_{v_1} \neq \emptyset \neq E_{v_2}$ (otherwise the structure is not well-formed).

Iterating an edge: Let e be an edge with $\nu(e) = (v_1, v_2, v_3)$. Then a fresh edge e' with $\nu(e') := (v_1, v_2, v_3)$ and $\kappa(e') := \kappa(e)$ may be inserted. That is, we construct the graph $\mathcal{G}^i := (V, E^i, \nu^i, \kappa)$ with $E^{(i)} := E \cup \{e'\}$ and $\nu^i := \nu \cup \{(e', (v_1, v_2, v_3))\}$.

Isomorphism: Two graphs which are isomorphic are implicitly identified.

A finite sequence $(\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_n)$ such that each $\mathcal{G}_{i+1}$ is derived from $\mathcal{G}_i$ by applying one of the rules above is a PROOF. We say that $\mathcal{G}_n$ can be DERIVED FROM $\mathcal{G}_1$ and write $\mathcal{G}_a \vdash \mathcal{G}_b$. Two graphs $\mathcal{G}_a, \mathcal{G}_b$ with $\mathcal{G}_a \vdash \mathcal{G}_b$ and $\mathcal{G}_b \vdash \mathcal{G}_a$ are said to be PROVABLY EQUIVALENT. The calculus is adequate, i.e. we have:

Theorem 1. *Two graphs $\mathcal{G}_a, \mathcal{G}_b$ satisfy $\mathcal{G}_a \vdash \mathcal{G}_b \Leftrightarrow \mathcal{G}_a \models \mathcal{G}_b$.*

5 Restricting the Graphs to Normal Forms

We have seen that triple sets directly correspond to graphs in NF (or in ANF). Thus the question arises whether we can use the results of the last sections, particularly the sound- and completeness of the calculus, for the system of RDF graphs in NF and the system of RDF graphs in ANF.

Clearly, the rules of the calculus are still sound if the system of graphs is syntactically restricted. But assume we restrict ourselves to graphs in NF: It is not allowed to split a vertex, as this yields a graph which is not in NF. Vice versa, it is not possible to apply the rule ‘merging two vertices’: We never find two different vertices which are identically labeled. Let us consider the following valid entailment between two RDF triple sets: $(o_1 \text{ } p \text{ } o_2) \models (o_1 \text{ } p \text{ } o_2), (\_1 \text{ } p \text{ } o_2)$. This entailment can directly translated to a valid entailment for graphs in NF or in ANF, using the mappings Φ_N or Φ_{AN} . But in none of the restricted systems,

we can find a formal proof within the given calculus. So for each restricted system, we have to alter the calculus to obtain completeness.

The rules ‘erasing an edge’ and ‘generalizing a label’ transform RDF graphs in ANF into RDF graphs in ANF again. A third rule is obtained as follows: By iterating an edge e and then splitting each of its three incident vertices, we can obtain a copy of e with *fresh* vertices. The combination of these rules is called *iterating and isolating the edge e* . This rule transforms RDF graphs in ANF into RDF graphs in ANF, too. We will write $\mathcal{G}_a \vdash_{an} \mathcal{G}_b$, if $\mathcal{G}_b$ can be derived from $\mathcal{G}_a$ with these three rules.

The rules ‘erasing an edge’ and ‘generalizing a label’ can be used for RDF graphs in NF as well. If a graph $\mathcal{G}_b$ is derived from $\mathcal{G}_a$ by firstly splitting a vertex v_1 into v_1 and a fresh vertex v_2 , and then by generalizing the label of v_2 , then we say that $\mathcal{G}_b$ is derived from $\mathcal{G}_a$ by SPLITTING AND GENERALIZING A VERTEX. These three rules transform RDF graphs in NF into RDF graphs in NF. We will write $\mathcal{G}_a \vdash_n \mathcal{G}_b$, if $\mathcal{G}_b$ can be derived from $\mathcal{G}_a$ with these three rules.

Theorem 2. *Let $\mathcal{G}_a$ and $\mathcal{G}_b$ be two RDF graphs with $\mathcal{G}_a \models \mathcal{G}_b$. If both graphs are in NF, we have $\mathcal{G}_a \vdash_n \mathcal{G}_b$. If both graphs are in ANF, we have $\mathcal{G}_a \vdash_{an} \mathcal{G}_b$.*

6 Outlook

This paper is only a first step for developing the underlying logic of the Semantic Web as mathematical DRS. The expressivity of RDF is rather weak. So, of course, it has to be investigated how well-known extensions of RDF can be developed as DRS. A paper which investigates how Description Logics, which are the underlying logics of the state-of-the-art SW-language OWL, is in preparation.

References

- [1] J. Hayes: A Graph Model for RDF. Diploma thesis, Technische Universität Darmstadt, Department of Computer Science, Germany, <http://purl.org/net/jhayes/rdfgraphmodel.html> (2004)
- [2] J. Hayes, C. Gutierrez: *Bipartite Graphs as Intermediate Model for RDF*. In: Proceedings of ISWC 2004, LNCS 3298, Springer, 2004.
- [3] P. Hayes: *RDF Semantics: W3C Recommendation*. 10 February 2004. <http://www.w3.org/TR/rdf-mt/>
- [4] P. Hayes, C. Menzel: *A Semantics for the Knowledge Interchange Format*. Proceedings of 2001 Workshop on the IEEE Standard Upper Ontology, August 2001. <http://reliant.tekknowledge.com/IJCAI01/HayesMenzel-SKIF-IJCAI2001.pdf>
- [5] G. Klyne, J.J. Carroll: *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C Recommendation. <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>
- [6] R. Kremer: *Visual Languages for Knowledge Representation*. Proc. of (KAW’98) Banff, Morgan Kaufmann, 1998.
- [7] J. H. Larkin H. A. Simon: *Why a diagram is (sometimes) worth ten thousand words*. Cognitive Science, 11, 65-99.
- [8] F. Manola, E. Miller: *RDF Primer*. <http://www.w3.org/TR/rdf-primer/>

A Framework for Defining and Verifying Clinical Guidelines: A Case Study on Cancer Screening

Federico Chesani¹, Pietro De Matteis², Paola Mello¹,
Marco Montali¹, and Sergio Storari³

¹ DEIS, University of Bologna, Bologna, Italy
{fchesani, pmello, mmontali}@deis.unibo.it

² NOEMALIFE, Bologna, Italy
pgdematteis@dianoema.it

³ ENDIF, University of Ferrara, Ferrara, Italy
strsrg@unife.it

Abstract. Medical guidelines are clinical behaviour recommendations used to help and support physicians in the definition of the most appropriate diagnosis and/or therapy within determinate clinical circumstances. Due to the intrinsic complexity of such guidelines, their application is not a trivial task; hence it is important to verify if health-care workers behave in a conform manner w.r.t. the intended model, and to evaluate how much their behaviour differs.

In this paper we present the GPROVE framework that we are developing within a regional project to describe medical guidelines in a visual way and to automatically perform the conformance verification.

1 Introduction

Medical guidelines are clinical behaviours recommendations used to help and support physicians in the definition of the most appropriate diagnosis and/or therapy within determinate clinical circumstances. These guidelines, usually represented as flow-charts, are intrinsically complex; moreover, their application in real cases becomes even more problematic. Hence, it is very important to verify if health-care workers applying them are conform with the intended models, and to evaluate how much their applications differ.

The sanitary organization of the Emilia Romagna region (Italy) uses these guidelines to describe and to manage cancer screening programs, in particular screening about cervical, breast and colorectal cancers. These guidelines are used to organize the screening, support test execution, analyze the conformance of the health-care workers behavior and evaluate how well the screening model “fits” the real screening application. Due to the management complexity of these screening programs, the Emilia Romagna sanitary organization is interested in software systems capable to support all these tasks.

In this paper we present the Guideline PRocess cOnformance VERification framework (GPROVE) that we are developing within the SPRING PRRITT regional project. Several software tools already address the guideline representation and execution issues but, to the best of our knowledge, no one deeply explores the aspect of conformance verification. Indeed, the SPRING PRRITT project aims to develop a system for supporting

health-care workers involved in the cancer screening programs, covering both management and conformance verification issues. To this purpose, the GPROVE framework proposes a new visual representation language and a way to integrate it within a formal framework, based on computational logic, originally developed in the context of SOCS European project [1].

2 Formalization of the Colorectal Cancer Screening Guideline

As a case study for exploiting GPROVE potentialities we choose the colorectal cancer screening guideline proposed by the sanitary organization of the Emilia Romagna region [2]. Colorectal cancer is a disease in which malignant (cancer) cells form in the tissues of the colon or the rectum. Conformance verification for this screening is useful to compute statistics about the screening process used by the board of the sanitary organization, to monitor the progress of the screening program and to eventually propose changes.

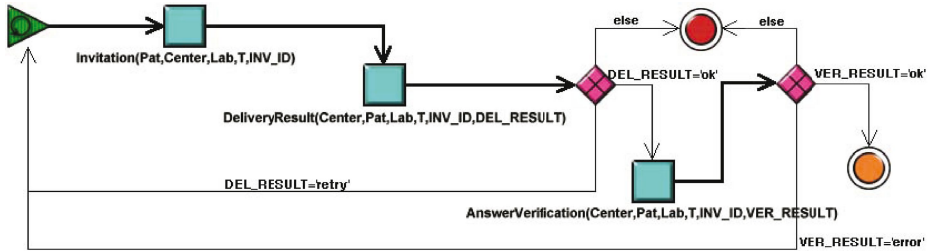


Fig. 1. A fragment of the screening guideline in GOSPEL: the invitation phase

The guideline describes the screening program as composed by six phases: (1) Screening Planning, (2) Invitation Management, (3) First Level Test with Fecal Occult Blood Test, (4) Second Level Test with Colonoscopy, (5) Second Level Test with Opaque Clisma and (6) Therapeutic Treatment. Thanks to the GOSpeL language we have developed a prototype representation of this guideline (the methodology and the language used are described in Section 3). Figure 1 shows, for example, the GOSpeL flow chart diagram of phase (1).

3 Architecture and Modules of the GPROVE Framework

GPROVE offers several functionalities that allow an easy graphical guideline representation and the automatic translation of such language to a formal one. Thanks to this formal representation of the guideline knowledge, it is possible to perform several analysis on the behavior of the subjects involved in the process described by the guideline. The framework architecture, shown in Figure 2, is composed by four modules: the GOSpeL Editor, the Translation Module, the Verification Module and the Event Mapping Module.

The GOSpeL Language and Editor. GOSpeL is a graphical language, inspired by flow charts, for the specification and representation of all the activities that belong to a process and their flow inside it. The GOSpeL representation of a guideline consists of two different parts: a *flow chart*, which models the process evolution, and an *ontology*, which describes at a fixed level of abstraction the application domain and gives a semantics to the diagram.

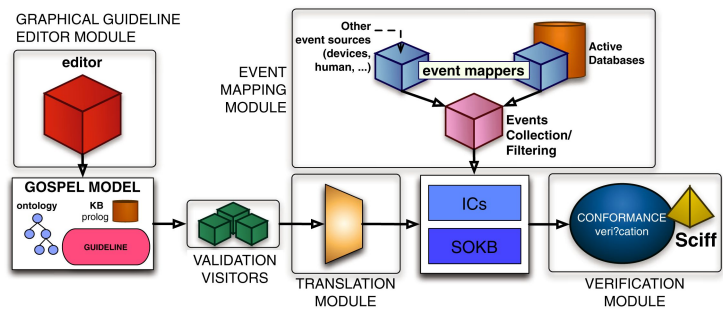


Fig. 2. Architecture and modules of the GPROVE framework

Like many other flow-chart languages, GOSpeL describes the guideline evolution using blocks and relations between blocks. These blocks are grouped into four families (as shown in Table 1): *activities*, blocks which represent guideline activities at the desired abstraction level; *gateways*, blocks that are used to manage the convergence (merge-block) and the divergence (split-block) of control flows; *start* blocks, start points of (sub)processes; *end* blocks, end points of (sub)processes. Due to the lack of space, we omit the detailed description of the GOSpeL language: the interested reader can refer to [3].

Table 1. GOSpeL elements

activities				gateways				start blocks		end blocks	
atomic activity	complex activity	iteration	while	exclusive choice	deferred choice	parallel fork	parallel join	start	cyclic start	return	end

In the example of the GOSpeL diagram used to represent the invitation management phase, shown in Figure 1, the *DeliveryResult(...)* action block expresses that after the invitation, the screening center waits for the invitation delivery result (*DEL_RESULT*) and if the delivery fails (e.g. due to a wrong address) the invitation is repeated while if the patient is emigrated or died the screening process is aborted. In order to know which path should be followed, variable *DEL_RESULT* is used. For example, the invitation

repetition case is represented by an arrow which connects the deferred choice to the cyclic start and is labeled by the logical condition $DEL\_RESULT = 'retry'$.

In the editing phase, a semantics is given to the model by means of a domain ontology. This ontology is mainly composed by two taxonomies: one is developed by knowledge experts to model activities of interest, whereas a second taxonomy describes domain's entities, namely *actors*, *objects* and *terms*. Each ontological activity is associated with one or more ontological entities that are involved in its execution. Each atomic activity block in the GOSpeL language is semantically specified by assigning it an ontological activity and a set of formal participant of the requested type.

Participants are introduced within *macroblocks* giving them a unique name; *macroblocks* define also precise visibility rules of participants. When specifying an activity, visible participants can be associated to the selected ontological activity. During the execution, each participant will be grounded to a concrete instance. E.g., the activity block *DeliveryResult* in Figure 1 is associated with the correspondent *DeliveryResult* ontological concept. Its formal participants, like *Pat* and *Center*, are associated with ontological entities too (e.g., *Patient* and *Place* respectively). Creation and management of ontologies is performed through the Protege [4] tool.

The Formal Language and the Translation Module. The formal language used in the framework is a simplified version of the one proposed by Alberti et al. in the SOCS European project for the specification and verification of interaction protocols (see [5] for a complete description). The ongoing social participants behaviour is represented by a set of (ground) facts called *events* and denoted by the functor **H**(that stands for “happened”). For example, $H(invitation(strsrg, center1, lab1, '10-02-06', 300), 7)$ represents the fact that *center1* has sent the invitation with ID 300 to *strsrg* at time 7.

Even if the participants behavior is unpredictable, once interaction protocols are defined we are able to determine what are the possible expectations about future events. Events expected to happen are indicated by the functor **E**. Expectations have the same format as events, but they will, typically, contain variables, to indicate that expected events are not completely specified. CLP constraints [6] can be imposed on variables to restrict their domain. For instance, $E(DeliveryResult(Center, Pat, Lab, T, INV\_ID, DEL\_RESULT), T_e) \wedge T_e \leq 15$ represents the expectation for *Center* to verify the delivery result of the invitation of *Pat* at a certain time T_e , whose value must be lower than 15 (i.e., a deadline has been imposed on T_e).

The way expectations are generated, given the happened events and the current expectations, is specified by means of *Social Integrity Constraints* (IC_S). An *IC* has the form of $body \rightarrow head$, expressing that when *body* becomes true then it is supposed that the events specified in *head* will happen. In this way, we are able to define protocols as sets of forward rules, relating events to expectations.

During the translation phase, atomic activity blocks are treated as observable and relevant events, whereas merge and split blocks determine how events are composed within an integrity constraint. Start and end blocks map to special events used to identify the initial and the final step of a (sub)process.

The translation algorithm operates in two steps: in the first one, it partitions the diagram in special sub-sets called *minimal windows*; in the second one, it translates each minimal window to an *IC*. Interested readers can refer to [3] for a detailed description of

the translation algorithm. Equation 1 has been generated automatically by the translation module, applied on the guideline of Figure 1. It states that, when a screening center *Center* verifies the invitation delivery status *DEL_RESULT*, then one among three next events are expected, depending on the value of *DEL_RESULT*.

$$\begin{aligned}
 & \mathbf{H}(\text{DeliveryResult}(\text{Center}, \text{Pat}, \text{Lab}, T, \text{INV\_ID}, \text{DEL\_RESULT}), T_{res}) \\
 \rightarrow & \mathbf{E}(\text{AnswerVerification}(\text{Center}, \text{Pat}, \text{Lab}, T, \text{INV\_ID}, \text{VER\_RESULT}), T_{ver}) \\
 & \wedge \text{DEL\_RESULT} = \text{'ok'} \wedge T_{ver} > T_{res} \\
 \vee & \text{E}(\text{start}(\text{invitation}), T_s) \wedge \text{DEL\_RESULT} = \text{'retry'} \wedge T_s > T_{res} \\
 \vee & \text{E}(\text{abort}(), T_a) \wedge \text{DEL\_RESULT} \neq \text{'ok'} \wedge \text{DEL\_RESULT} \neq \text{'retry'} \wedge T_a > T_{res}
 \end{aligned} \tag{1}$$

The Verification Module. The conformance verification is performed by means of the operational counterpart of IC_S , i.e. the SCIFF abductive proof procedure [7]. Given the partial or the complete history of a specific execution (i.e., the set of already happened events recorded in a event log), SCIFF generates expectations about participants behaviour so as to comply with IC_S . A distinctive feature of SCIFF is its ability to check that the generated expectations are *fulfilled* by the actual participants behaviour (i.e., that events expected to happen have actually happened). If a participant does not behave as expected w.r.t. the model, the proof procedure detects such discrepancy and raises a violation.

Two kinds of guideline conformance violations are detected by the Verification Module: the first violation type happens when SCIFF does not find in the log an event that was expected ($\mathbf{E}(\text{event})$ without the relative $\mathbf{H}(\text{event})$); the second violation type is detected when it finds an event that was not expected ($\mathbf{H}(\text{event})$ without the relative $\mathbf{E}(\text{event})$): if an event is not explicitly expected, then it is automatically considered as prohibited.

At the moment, we are interested in verifying “a posteriori” if the event log of a guideline process case is conformant w.r.t. the model. However, since SCIFF is able to perform both off-line and run-time verification, we are also investigating the possibility to check during the guideline application if participants behave in a conform manner.

Event Mapping Module. The proposed event mapping module is capable to map ontological activities to a concrete DBMS and interact with it in order to extract the corresponding events and create an event log. Events are actively sent to the event mapping module by the DBMS (i.e. by triggers) or can be acquired via query statements.

4 Conclusions and Future Works

In this paper we have described a new framework, named GPROVE, for specifying medical guidelines and for verifying the conformance of the guideline execution w.r.t the specification. This framework is composed by four modules, that allow the user to graphically specify the guidelines (by means of the GOSpeL language and editor), to translate the specification into a formal language, to capture the events associated with the guideline execution and to verify the conformance of such executions w.r.t. the guideline. GPROVE is currently developed in the context of an Italian regional project

on cancer screening management. The feasibility of the approach has been exploited on the real scenario of the colorectal cancer screening.

The literature proposes many systems related to the GPROVE framework. GLARE [8] and PROforma [9], to cite some, are powerful tools for supporting both representation and execution of medical guidelines. GPROVE, at the moment, is not integrated with an execution engine. However, it addresses the conformance verification issue that, as far as we know, neither GLARE nor PROforma tackle (indeed, GLARE deals only with time constraint conformance). Hence GPROVE can be considered a complementary approach w.r.t the ones proposed by GLARE and PROforma. Moreover, GPROVE can be used to model processes in different application domains, by means of different ontologies.

The conformance verification issue is addressed in [10], where the authors define the conformance verification of a process model on a event log as an analysis that involves two aspects: underfitting and overfitting. In our framework, the SCIFF proof procedure identifies both of them even if with a limited set of conformance statistics that will be improved soon.

In the future we plan to complete the development of the GPROVE modules and to extend the expressive power of GOSpeL, maintaining at the same time a valid mapping to the formalism. We will also deal with conformance verification at execution time.

Acknowledgments. This work has been partially supported by NOEMALIFE under the regional PRRITT project “Screening PRotocol INtelligent Government (SPRING)” and by the PRIN 2005 project “Specification and verification of agent interaction protocols”.

References

1. Societies Of Computees (SOCS), IST-2001-32530, <http://lia.deis.unibo.it/research/socs/> (2006)
2. Emilia Romagna, Colorectal cancer screening in the Emilia Romagna region of Italy, <http://www.saluter.it/colon/documentazione.htm> (2006)
3. Chesani, F., Ciampolini, A., Mello, P., Montali, M., Storari, S.: Testing guidelines conformance by translating a graphical language to computational logic. In: To appear in the proceedings of the ECAI 2006 workshop on AI techniques in healthcare: evidence-based guidelines and protocols. (2006)
4. Protege, <http://stanford.protege.org> (2006)
5. Alberti, M., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Specification and verification of agent interactions using social integrity constraints. *Electronic Notes in Theoretical Computer Science* **85**(2) (2003)
6. Jaffar, J., Maher, M.: Constraint logic programming: a survey. *J. of Logic Programming* **19-20** (1994) 503–582
7. The SCIFF Abductive Proof Procedure, <http://lia.deis.unibo.it/research/sciff/> (2006)
8. Terenziani, P., Montani, S., Bottrighi, A., Torchio, M., Molino, G., Correndo, G.: The glare approach to clinical guidelines: main features. *Stud. Health Tech. Inf.* **101** (2004) 162–166
9. Fox, J., Johns, N., Rahmzadeh, A.: Disseminating medical knowledge-the proforma approach. *Artificial Intelligence in Medicine* **14** (1998) 157–181
10. van der Aalst, W.M.P.: Business alignment: Using process mining as a tool for delta analysis and conformance testing. To appear in *Requirements Engineering Journal* (2006)

Preferred Generalized Answers for Inconsistent Databases

L. Caroprese, S. Greco, I. Trubitsyna, and E. Zumpano

DEIS, Univ. della Calabria, 87030 Rende, Italy
{lcaroprese, greco, irina, zumpano}@deis.unical.it

Abstract. The aim of this paper consists in investigating the problem of managing inconsistent databases, i.e. databases violating integrity constraints. A flurry of research on this topic has shown that the presence of inconsistent data can be resolved by “repairing” the database, i.e. by providing a computational mechanism that ensures obtaining consistent “scenarios” of the information or by consistently answer to queries posed on an inconsistent set of data. This paper considers preferences among repairs and possible answers by introducing a partial order among them on the basis of some preference criteria. Moreover, the paper also extends the notion of preferred consistent answer by extracting from a set of preferred repaired database, the maximal consistent overlapping portion of the information, i.e. the information supported by each preferred repaired database.

1 Introduction

A flurry of research has investigated the problems related to the management of inconsistent databases. Focusing on the various approaches, proposed in the literature [1,5,6,7,10,11,14], inconsistencies are generally solved by “repairing” the database, i.e. by providing a computational mechanism that ensures obtaining consistent “scenarios” of the information (repairs) in an inconsistent environment or to consistently answer to queries posed on an inconsistent set of data. The computation of repairs is based on the insertion and deletion of tuples so that the resulting database satisfies all constraints, whereas the computation of consistent answers is based on the identification of tuples satisfying integrity constraints and matching the goal [2,3,7,13]. If alternative repairs are allowed, the repairing process can be driven by specifying the preferred update operations, i.e. those actions that lead to obtain preferred repaired databases. To this aim, in this paper we introduce a flexible mechanism that allows specifying preference criteria, i.e. desiderata on how to update the inconsistent database in order to make it consistent, so that selecting among a set of repairs the preferred ones, i.e. those better conforming to the specified criteria.

It’s well known that in the presence of integrity constraints, it is often likely to have more (preferred) repairs. In such a case, a deterministic consistent semantics is, classically, given by selecting as true information that present in all (preferred)

repaired database, i.e the set of atoms belonging to the intersection of (preferred) repaired databases. A more flexible semantics can be given by selecting as true information that supported by all (preferred) repaired databases, i.e. the set of atoms which maximizes the information shared by the (preferred) repaired databases. The following example clarifies the benefit of this purpose.

Example 1. Consider the database consisting of the unique binary relation $Teaches(Course, Professor)$ and the following integrity constraint $(\forall (X, Y, Z)) [Teaches(X, Y), Teaches(X, Z) \supset Y = Z]$ stating that a course cannot be taught by different professors. Suppose to have the database instance $D = \{Teaches(c_1, p_1), Teaches(c_2, p_2), Teaches(c_2, p_3)\}$. The two alternative repaired databases are: $R_1(D) = \{Teaches(c_1, p_1), Teaches(c_2, p_3)\}$ (obtained by deleting the tuple $Teaches(c_2, p_2)$) and $R_2(D) = \{Teaches(c_1, p_1), Teaches(c_2, p_2)\}$ (obtained by deleting the tuple $Teaches(c_2, p_3)$). If no preference criteria drives the repairing process, then both $R_1(D)$ and $R_2(D)$ are equally preferred. Considering the standard semantics the set of true atoms consists of the unique tuple $Teaches(c_1, p_1)$, whereas, intuitively, the information supported by the repaired databases also contains the tuple $Teaches(c_2, \perp)$, stating that there exists a course c_2 even though the professor teaching that course is unknown. In fact c_2 is taught by p_3 in $R_1(D)$ and by p_2 in $R_2(D)$, thus we don't know "exactly" who is the professor teaching c_2 . $\square$

2 Preferred Repairs and Consistent Answers

Familiarity is assumed with relational database theory [12]. In this section the definition of consistent database and repair is first recalled and, then, a computational mechanism is presented that ensures selecting preferred repairs and preferred answers for inconsistent database.

Definition 1. *Consistent database.* Given a database D over a database schema DS and a set of integrity constraints $\mathcal{IC}$ on D , D is *consistent* if $D \models \mathcal{IC}$, i.e. if all integrity constraints in $\mathcal{IC}$ are satisfied by D , otherwise it is *inconsistent*. $\square$

Definition 2. *Repair [2].* Given a (possibly inconsistent) database D , a *repair* for D is a pair of sets of atoms (R^+, R^-) such that 1) $R^+ \cap R^- = \emptyset$, 2) $D \cup R^+ - R^- \models \mathcal{IC}$ and 3) there is no pair $(S^+, S^-) \neq (R^+, R^-)$ such that $S^+ \subseteq R^+$, $S^- \subseteq R^-$ and $D \cup S^+ - S^- \models \mathcal{IC}$. The database $D \cup R^+ - R^-$ will be called the *repaired database*. $\square$

Thus, repaired databases are consistent databases which are derived from the source database by means of a minimal set of insertion and deletion of tuples. Given a repair R , R^+ denotes the set of tuples which will be added to the database, whereas R^- denotes the set of tuples which will be deleted from the database. In the following, for a given repair R and a database D , $R(D) = D \cup R^+ - R^-$ denotes the application of R to D . The set of all repairs for a database D and integrity constraints $\mathcal{IC}$ is denoted by $\mathbf{R}(D, \mathcal{IC})$.

In [8] a polynomial function for expressing preference criteria, has been introduced. It specifies a partial order among repairs, so that allowing the evaluation of the goodness of a repair. In particular, in such a context a preferred repair is a repair minimal with respect to the partial order.

Definition 3. *Preferred Repair.* Given a (possibly inconsistent) database D over a fixed schema $\mathcal{DS}$, and a polynomial function $f : (\mathbf{D}, \mathbf{D}) \times \mathbf{D} \rightarrow \mathbb{R}^+$. Given two repairs R_1 and R_2 of D , R_1 is *preferable to* R_2 w.r.t. f , written $R_1 \ll_f R_2$, if $f(R_1, D) \leq f(R_2, D)$. R_1 is *preferred to* R_2 w.r.t. f , written $R_1 <_f R_2$, if $f(R_1, D) < f(R_2, D)$. A repair R for D is said to be *preferred w.r.t. f* if there is no repair R' for D such that $R' <_f R$. A repaired database $D' = D \cup R^+ - R^-$ is said to be a *preferred database* if $R = (R^+, R^-)$ is a preferred repair. $\square$

The above function f will be called (*repair*) *evaluation function* as it is used to evaluate a repair R with respect to a database D . A preferred database minimizes the value of the evaluation function f applied to the source database and repairs. Observe that, in the above definition, $\mathbb{R}^+$ denotes the set of non negative real numbers, $\mathbf{D}$ denotes the domain of all possible database instances, whereas $(\mathbf{D}, \mathbf{D})$ denotes the domain of all possible database updates. Thus the evaluation function f can be used to measure any possible modification of the input databases and not only to measure repairs, i.e. modifications which make the database consistent. For the sake of simplicity, we will only consider functions which minimize the cardinality of a set.

Given a database D , a set of integrity constraints $\mathcal{IC}$ and an evaluation function f , we denote with $\mathbf{R}(D, \mathcal{IC}, f)$ the set of preferred repairs for D .

Consider Example 1 and suppose $f(R, D) = |R^-|$, computing the number of deleted atoms, then $\mathbf{R}(D, \mathcal{IC}, f) = \{R_1, R_2\}$.

Moreover, we denote with f_0 any constant evaluation function (e.g. $f_0(R, D) = 0$, the function returning the value 0). $\mathbf{R}(D, \mathcal{IC}, f_0)$ denotes the set of all repairs for D as no preference is introduced.

Definition 4. Given a database D , a set of integrity constraints $\mathcal{IC}$, and an evaluation function f , an atom A is *true* (resp. *false*) w.r.t. $\mathcal{IC}$ and f , if A belongs to all preferred repaired databases (resp. there is no preferred repaired database containing A). Atoms which are neither true nor false are *undefined*. $\square$

Thus, true atoms appear in all preferred repaired databases, whereas undefined atoms appear in a proper subset of preferred repaired databases.

Definition 5. *Consistent Answer.* Given a database D over a relational schema $\mathcal{DS}$ with integrity constraints $\mathcal{IC}$, an evaluation function f and a query Q , the consistent answer to Q , denoted $Q(D, \mathcal{IC}, f)$ is given by the set $\{Q(R(D)) | R \in \mathbf{R}(D, \mathcal{IC}, f)\}$. $\square$

The deterministic semantics of queries is given by considering the set of repaired databases under brave and cautious reasoning.

Definition 6. *Deterministic Answer.* Given a database D over a relational schema $\mathcal{DS}$ with integrity constraints $\mathcal{IC}$, an evaluation function f and a query Q , the *deterministic answer* to the query Q over a database D is

- $\bigcup_{R \in \mathbf{R}(D, \mathcal{IC}, f)} Q(R(D))$, under brave reasoning, and
- $\bigcap_{R \in \mathbf{R}(D, \mathcal{IC}, f)} Q(R(D))$, under cautious reasoning. $\square$

The (deterministic) preferred consistent answer consists of true and undefined atoms.

Definition 7. *Deterministic Preferred Answer.* Given a database D over a relational schema $\mathcal{DS}$ with integrity constraints $\mathcal{IC}$, an evaluation function f and a query Q , the (deterministic) preferred answer to the query Q over the database D is a couple $\langle Q^+(D, \mathcal{IC}, f), Q^u(D, \mathcal{IC}, f) \rangle$, where

- $Q^+(D, \mathcal{IC}, f) = \bigcap_{R \in \mathbf{R}(D, \mathcal{IC}, f)} Q(R(D))$ denotes the set of *true* atoms, and
- $Q^u(D, \mathcal{IC}, f) = \bigcup_{R \in \mathbf{R}(D, \mathcal{IC}, f)} Q(R(D)) - Q^+(D, \mathcal{IC}, f)$ denotes the set of *undefined* atoms. $\square$

The set of false atoms consisting of all atoms which are neither true nor undefined is $Q^-(D, \mathcal{IC}, f) = \mathbf{D} - Q^+(D, \mathcal{IC}, f) - Q^u(D, \mathcal{IC}, f) = \mathbf{D} - \bigcup_{R \in \mathbf{R}(D, \mathcal{IC}, f)} Q(R(D))$.

Theorem 1. *Given a database D over a relational schema $\mathcal{DS}$ with integrity constraints $\mathcal{IC}$, an evaluation function f and a query Q , then*

1. $\mathbf{R}(D, \mathcal{IC}, f) \subseteq \mathbf{R}(D, \mathcal{IC}, f_0)$
2. $Q(D, \mathcal{IC}, f_0) \subseteq Q(D, \mathcal{IC}, f)$ $\square$

3 Preferred Generalized Answers

In this section we extend the notion of deterministic preferred consistent answer by also allowing the presence of partially defined atoms, i.e. atoms with “unknown” values due to the presence of tuples in different repaired databases which disagree on the value of one or more attributes. In other words, while in the previous section we have assigned to every ground atom a truth value which can be *true*, *false* or *undefined*, in this section we extract the maximal consistent portion of the information on which the preferred repaired databases agree. To this aim, we introduce in the following a binary relationship for comparing two ground atoms or two sets of ground atoms.

Definition 8. Given two ground atoms $A = p(t_1, \dots, t_n)$ and $B = p(u_1, \dots, u_n)$ we say that A *subsumes* B (written $A \vdash B$) if $\forall i$ either $t_i = u_i$ or $t_i = \perp$. Given two sets of ground atoms S_1 and S_2 , S_1 *subsumes* S_2 (written $S_1 \vdash S_2$) if $\forall B \in S_2 \exists A \in S_1$ s.t. $A \vdash B$ and $\forall A \in S_1 \exists B \in S_2$ s.t. $A \vdash B$. $\square$

Example 2. The set $\{p(a, \perp), p(\perp, b)\}$ subsumes the set $\{p(a, d), p(c, b)\}$. $\square$

Thus, given two ground atoms A and B , A subsumes B if each term in A is equal or less specific than the correspondent term in B . Note that the relation $\vdash$ is transitive (if $A \vdash B$ and $B \vdash C$, then $A \vdash C$) and antisymmetric (if $A \vdash B$, then $B \nvdash A$).

Definition 9. Given a set of sets of ground atoms S and a set of ground atoms T we say that: (i) T *generalizes* S (written $T \models S$) if for each $S_i \in S$ is $T \vdash S_i$; (ii) T is the *minimal generalization* of S if $T \models S$ and there is no set $U \neq T$ such that $T \vdash U \models S$. $\square$

Example 3. The minimal generalization of the set of sets $\{ \{p(a, b)\}, \{p(a, d), p(c, b)\} \}$ is $\{p(a, \perp), p(\perp, b)\}$. $\square$

Given a set sets of (preferred) repaired database $\mathbf{R}$, the *generalized repaired database*, denoted as $\widehat{R(D)}$, is the minimal generalization of $\mathbf{R}$.

Example 4. Consider the Example 1. The generalized repaired database for $\{ R_1(D), R_2(D) \}$ is $\widehat{R(D)} = \{ Teaches(c_1, p_1), Teaches(c_2, \perp) \}$. $\square$

Definition 10. Let D be a database over a relational schema $\mathcal{DS}$, $\mathcal{IC}$ a set of integrity constraints, $\widehat{R(D)}$ the generalized repaired database and f an evaluation function, an atom A is *true* (resp. *false*) w.r.t. $\mathcal{IC}$ and f , if $A \in \widehat{R(D)}$ (resp. $A \notin \widehat{R(D)}$). The atoms which are neither true nor false are *undefined*. $\square$

Thus, true atoms are subsumed by all preferred repaired databases, whereas undefined atoms appear in a proper subset of preferred repaired databases.

Definition 11. *Deterministic Generalized Answer.* Let D be a database, $\mathcal{IC}$ a set of integrity constraints, f an evaluating function, Q a query, and $\widehat{R(D)}$ the generalized repaired database, then the deterministic generalized answer for the query Q , over the database D , is a couple $\langle \widehat{Q}^+(D, \mathcal{IC}, f), \widehat{Q}^u(D, \mathcal{IC}, f) \rangle$, where

- $\widehat{Q}^+(D, \mathcal{IC}, f) = Q(\widehat{R(D)})$ denotes the set of *true* atoms, and
- $\widehat{Q}^u(D, \mathcal{IC}, f) = \bigcup_{R \in \mathbf{R}(D, \mathcal{IC}, f)} Q(R(D)) - \widehat{Q}^+(D, \mathcal{IC}, f)$ denotes the set of *undefined* atoms. $\square$

Example 5. Consider the Example 1. If no preference criteria is given, then both $R_1(D)$ and $R_2(D)$ are preferred repaired database. Suppose to have the query $Q = (Teaches, \emptyset)$. The minimal generalization of $\{R_1(D), R_2(D)\}$ is $\{ Teaches(c_1, p_1), Teaches(c_2, \perp) \}$, thus the deterministic generalized preferred consistent answer for the query Q , over the database D is a couple $\langle \widehat{Q}^+(D, \mathcal{IC}, f_0), \widehat{Q}^u(D, \mathcal{IC}, f_0) \rangle$, where

- $\widehat{Q}^+(D, \mathcal{IC}, f_0) = \{ Teaches(c_1, p_1), Teaches(c_2, \perp) \}$, that is we know that c_2 is a course, but is unknown the professor teaching that course;
- $\widehat{Q}^u(D, \mathcal{IC}, f_0) = \{ Teaches(c_2, p_2), Teaches(c_2, p_3) \} = Q^u(D, \mathcal{IC}, f_0)$, i.e. no change occurs to the set of undefined atoms. $\square$

Observe that the introduction of partially defined tuples generalizes the knowledge of undefined atoms. In fact, as shown in the previous example for the database of Example 1, we have derived that the facts $Teaches(c_2, p_2)$ and $Teaches(c_2, p_3)$ are undefined, but by using partially defined atoms, we add the knowledge that c_2 is a course.

Theorem 2. *Given a database D , a set of integrity constraint $\mathcal{IC}$, an evaluation function f and a query Q , then $Q^+(D, \mathcal{IC}, f) \subseteq \widehat{Q^+}(D, \mathcal{IC}, f)$.* $\square$

The above theorem states that by considering generalized answer we enlarges the set of true atoms in the answer as for a given set of (preferred) repaired databases $\mathbf{R}$ with generalized repaired database $\widehat{R(D)}$, atoms belonging to all sets in $\mathbf{R}$ also belong to $\widehat{R(D)}$.

Proposition 1. $\bigcap_{R \in \mathbf{R}(D, \mathcal{IC}, f)} R(D) \subseteq \widehat{R(D)}$. $\square$

Conclusions. In this paper a flexible mechanism that allows selecting among a set of repairs and possible answers those better conforming to specified preferences has been proposed. The problem related to the extraction of the maximal consistent information subsumed by a set of preferred database repairs is investigated; and the notion of preferred consistent answer is extended by allowing the presence of partially defined atoms, i.e. atoms with “unknown” values due to the presence of tuples in different (preferred) repaired databases which disagree on the value of one or more attributes.

References

1. Argaval, S., Keller, A. M., Wiederhold, G., Saraswat, K., Flexible Relation: an Approach for Integrating Data from Multiple, Possibly Inconsistent Databases. *ICDE*, pp. 495–504, 1995.
2. Arenas, M., Bertossi, L., Chomicki, J., Consistent query Answers in inconsistent databases. *PODS*, pp. 68–79, 1999.
3. Arenas, M., Bertossi, L., Chomicki, J., Specifying and Querying Database repairs using Logic Programs with Exceptions. *FQAS*, pp. 27–01, 2000.
4. Baral, C., Kraus, S., Minker, J., Combining Multiple Knowledge Bases. *TKDE*, 3(2), pp. 208–220, 1991.
5. Bry, F., Query Answering in Information System with Integrity Constraints, *IICIS*, pp. 113–130, 1997.
6. Dung, P. M., Integrating Data from Possibly Inconsistent Databases. *CoopIS*, pp. 58–65, 1996.
7. Greco, G., Greco, S., Zumpano, E., A logic programming approach to the integration, repairing and querying of inconsistent databases. *ICLP*, pp. 348–364, 2001.
8. Greco, S., Sirangelo, C., Trubitsyna, I., Zumpano, E., Preferred Repairs for Inconsistent Databases. *IDEAS*, pp. 202–211, 2003.
9. Grant, J., Subrahmanian, V. S., Reasoning in Inconsistent Knowledge Bases, *TKDE*, 7(1): 177–189, 1995.
10. Lin, J., A Semantics for Reasoning Consistently in the Presence of Inconsistency. *AI*, 86(1–2):75–95, 1996.
11. Subrahmanian, V. S., Amalgamating Knowledge Bases. *ACM TODS*, 19(2), pp. 291–331, 1994.
12. Ullman, J. K., *Principles of Database and Knowledge-Base Systems*, Vol. 1, Computer Science Press, 1988.
13. Wijnen, J., Condensed representation of database repairs for consistent query answering, *ICDT*, pp. 378–393, 2003.
14. Yan, L.L., Ozsu, M.T., Conflict Tolerant Queries in Aurora. *CoopIS*, pp. 279–290, 1999.

Representation Interest Point Using Empirical Mode Decomposition and Independent Components Analysis*

Dongfeng Han¹, Wenhui Li¹, Xiaosuo Lu¹, Yi Wang¹, and Ming Li²

¹ College of Computer Science and Technology, Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, Jilin University, Changchun, 130012, P.R. China
handongfeng@gmail.com

² JiLin Province Economic Information Center

Abstract. This paper presents a new interest point descriptors representation method based on empirical mode decomposition (EMD) and independent components analysis (ICA). The proposed algorithm first finds the characteristic scale and the location of the interest points using Harris-Laplacian interest point detector. We then apply the Hilbert transform to each component and get the amplitude and the instantaneous frequency as the feature vectors. Then independent components analysis is used to model the image subspace and reduces the dimension of the feature vectors. The aim of this algorithm is to find a meaningful image subspace and more compact descriptors. Combination the proposed descriptors with an effective interest point detector, the proposed algorithm has a more accurate matching rate besides the robustness towards image deformations.

1 Introduction

The problem of selection image interest points for computer vision problems such as matching [1], indexing [2], retrieval [3] and recognition [4, 5] has been extensively studied over the last thirty years, and many good schemes have been advanced. Just as described in [4], the common steps include:

1. Interest point detection and location: The common approach is LOG method proposed by Low. This procedure means computing searches over all scales and locations. At each candidate location, interest points are selected and the location and scale are determined.
2. Orientation assignment: The main orientation of each interest point is calculated. This can guarantee all interest points are in their canonical direction.
3. Interest points descriptors: The final step is to compute the descriptors of the interest points. The region around each interest point is transformed into a representation that allows for significant level of local distortion.

The common approaches described above always first detect features and then compute a set of descriptors for these features. How to compute the descriptors is a

* This work has been supported by NDSF Project 60573182, 69883004 and 50338030.

essential issue to the whole problem. Some good methods have been proposed (They will be described in section 1.1) to applied in different situations. In this paper, we present a new representation for descriptors. This method is based on empirical mode decomposition (EMD) and independent components analysis (ICA). The success of this method is based on an accurate interest point detector and a compact representation for point descriptors.

1.1 Relative Work on Descriptors

Many different approaches have been developed for describing local image. Cross-correlation can be used to compute the similarity between any two different descriptors. However, there is a contradiction between describing capacity and the dimension of the descriptors. Up to days, many algorithms exist, including ‘corners’ [6], parametric image models [7], local phase congruency [8], and morphology [9]. In [10], a more expressive representation is introduced by Johnson and Hebert for 3D object recognition. Their representation is a histogram of the relative position in the neighborhood of the interest point. In [11], Zabih and Woodfill have proposed a method robust to illumination changes. Their method relies on histogram of ordering and reciprocal relation between pixel intensities. The disadvantage of this method is that the dimension of the descriptors is too high to build practical descriptors. One of the most popular is SIFT method developed by Low [4]. The SIFT algorithm builds descriptors for each interest point based on a patch of pixel in its local neighborhood. The descriptors used in SIFT are computed by sampling the magnitudes and orientations of the image gradient in each patch center over the interest point, and building smoothed orientation histograms to capture the important aspects of the patch. A 4×4 array of the histograms, each with 8 orientation patch, capture the structure of the patch. This 128-element vector is normalized to unit length and remove the elements with small values.

Mikolajczyk in [12] compares shape context [13], steerable filters [14], PCA-SIFT [15], differential invariants [16], spin images [17], SIFT [4], complex filters [18], moment invariants [19], and cross-correlation for different types of interest regions and finds the SIFT based descriptors perform best. However, we find the features of SIFT can’t describe the local structure exactly and the dimension is too high to meet the real-time demand. Our goal is to find the effective descriptors which are more compact, faster and more accurate than the standard SIFT descriptors.

In sections 2, we will describe the interest point detector used in our algorithm. In section 3, the detail of our approach will be presented. Experiments and comparison will be given in section 4. In the end, we will give the conclusions and the future work in section 5.

2 Harris-Laplacian Interest Point Detector

In [20], authors evaluate some interest point detectors and point out that Harris detector [21] has an excellent performance compare to other methods. However, Harris detector is not invariant to scale changes. So we should first find the characteristic

scale for a given point. Then using Harris detector we can determine whether the point is an interest point.

The Harris detector combines the trace and the determinant of the auto-correlation matrix. The auto-correlation matrix is defined by:

$$M(x) = \sigma_D \cdot G(\sigma_I) * \begin{pmatrix} I_x^2(x, \sigma_D) & I_{x,y}(x, \sigma_D) \\ I_{x,y}(x, \sigma_D) & I_y^2(x, \sigma_D) \end{pmatrix} \quad (1)$$

Where σ_I is the integration scale, σ_D is the differentiation scale and I_α is the derivative computed in the α direction. So the point response is defined by:

$$R = \det M - k(\text{trace} M)^2 \quad (2)$$

Where $k = 0.04$.

The Harris-Laplacian detector in [22] uses the scale-adaptable Harris function Eq. (1) and Eq. (2) to locate the interest points in the scale-space. Then it selects the points for which the Laplacian-of-Gaussian (LOG) attains maximum over scales. The method consists of three stages: (i) an iterative selection of scale; (ii) interest points location; (iii) main orientation assignment. First, we build the scale-space representation using Harris function and detect local maxima at each scale. We reject the points for which the Laplacian attains no extremum or the response is below a threshold.

3 Our Algorithm

3.1 Our Contributions

Using Harris-Laplacian detector, we can get some interest points associated with characteristic scale. The next step is to give the representation for the interest points. How to effectively describe the local structure is a very important to the final performance. The standard SIFT descriptors perform best described in [11]. However, the construction of SIFT descriptors is complicated and the feature dimension is high. Our main contributions is to use EMD to get a new feature vectors and use ICA to get the high-level description from low-level image features with low dimension. It is the first time to introduce EMD and ICA to the matching problem. Experiments show that our algorithm can get a better result than SIFT method.

3.2 EMD Based Feature Vectors

Recently, Huang et al. [23] have introduced the empirical mode decomposition (EMD) method for analyzing data. Empirical mode decomposition (EMD) is a general nonlinear, non-stationary signal processing method. So it is very suit for describing local structure. The major advantage of the EMD is that the basis functions are derived from the signal itself. Hence, the analysis is adaptive, in contrast to the wavelet method where the basis functions are fixed. The central idea of EMD is to

decompose a time series into a finite and often small number of intrinsic mode functions (IMFs). Given a signal $x(t)$, the EMD can be represented as:

$$x(t) = \sum_{i=1}^N IMF_i(t) + r(t) \quad (3)$$

Where N is the number of IMFs and $r(t)$ is the residue of the signal.

An IMF must fulfil two requirements: (i) the number of extrema and the number of zero crossings are either equal or differ at most by one; (ii) at any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero. Thus, locally, each IMF contains higher frequency oscillations than the one just extracted before. So we just extract two IMFs that can represent the character of the signal enough.

Once obtaining the IMFs, we can apply Hilbert transform to each IMF and compute the amplitude and the frequency of each element. The data can express in the following form:

$$X(t) \approx \sum_{i=1}^2 a_i(t) \exp\left(j \int \omega_i(t) dt\right) \quad (4)$$

Form (4), we can get the amplitude and the instantaneous main orientation frequency of each element. We also extract the instantaneous frequency perpendicular to the main orientation. We do EMD on the data and get the first two IMFs which can represent the data enough. So we arrange the two amplitudes and four instantaneous frequencies for all the elements as the feature vectors. The feature vectors reflect the structure of the component. In Fig.1, the EMD on real data is given.

3.3 Independent Components Analysis

In image understanding, it is often desirable to represent an image in a subspace to reveal some of intrinsic characteristics of the data and reduce the dimension, where linear representation is often used due to its computational and analytical properties. Independent components analysis (ICA) [24] has such property. Basically, one attempts to represent each image patch as a linear combination of ‘basis’ patches, such that the mixing coefficients are as sparse as possible. In other words,

$$X = AS = \sum_{i=1}^n a_i s_i \quad (5)$$

Where X is the image patch, a_i is the basis images. ICA attempts to seek mutually independent components s_i . Most of the s_i will be close to zero and only a few will have significantly non-zero values. Once obtaining A , we can compute its inverse, say W , and obtain the independent components by:

$$S = WX = \sum_i^m w_i x_i \quad (6)$$

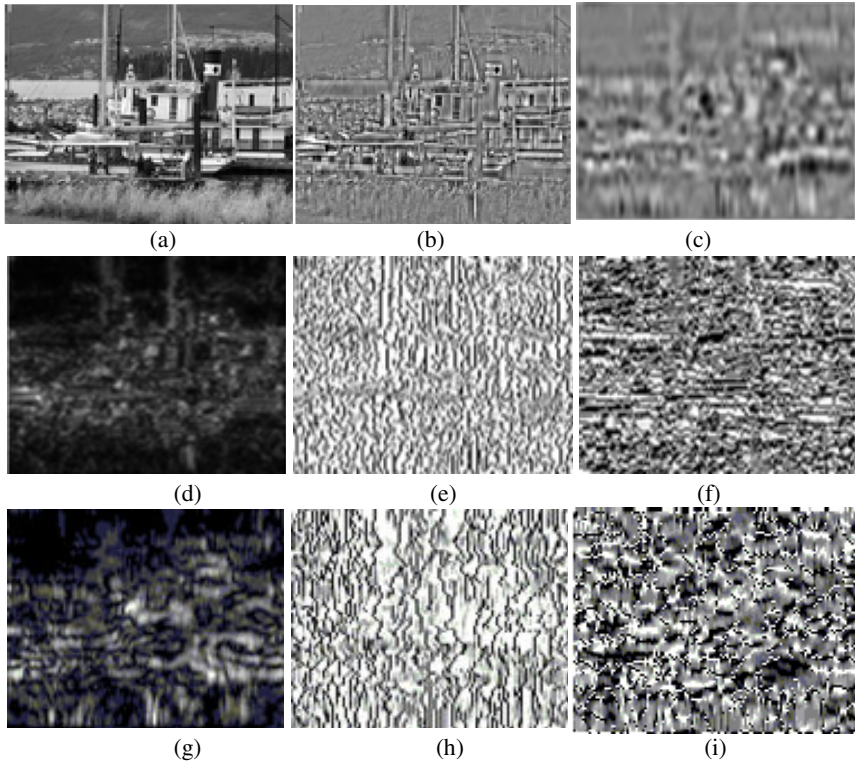


Fig. 1. Different parts of EMD. (a) Original image. (b) IMF_1 . (c) IMF_2 . (d) Amplitude of IMF_1 . (e) Instantaneous frequency of the main orientation for IMF_1 . (f) Instantaneous frequency of the orthogonal to the main orientation for IMF_1 . (g) Amplitude of IMF_2 . (h) Instantaneous frequency of the main orientation for IMF_2 . (i) Instantaneous frequency of the orthogonal to the main orientation for IMF_2 .

3.4 The Proposed Descriptors

We first describe the input vectors used in training ICA. In order to make the patches less sensitive to the exact feature location and noise effect, we sample the pixels at a lower frequency. We extract a 64×64 patch centered over the interest point and perform the Haar wavelet transform on the patch. We perform EMD on the LL sub-band (32×32). Each element has six features corresponding to 2 amplitudes and 4 instantaneous frequencies as described in section 3.2. So the length of the feature vectors is $6 \times 32 \times 32$. We normalize the feature vectors to unit magnitude to reduce the effects of illumination change.

Using Harris-Laplacian point detector, we extract 10,000 interest points from a common image database and create the input vectors for each interest point as described above. The whole input vectors are used to train ICA and the transformation matrix W in Eq. (5) can be computed.

ICA can linearly project high dimension vectors onto a low dimension feature space. Further more, it has a more meaningful visual property compared to PCA. Thus, the EMD and ICA based feature descriptors include following steps:

Step1. Extracting a 64×64 patch at the characteristic scale, centered over the interest point;

Step2. Rotating to its main orientation to a canonical direction;

Step3. Performing the Haar wavelet transform and getting LL sub-band;

Step4. Performing EMD on the LL sub-band and getting the first two IMFs. Then obtaining the amplitudes and the instantaneous frequencies;

Step5. Arranging the feature vectors and normalizing it to unit magnitude;

Step6. Using Eq. (6) to find the feature space for the $6 \times 32 \times 32$ input vectors;

The reduced feature space is the final interest point descriptors. In experiments, the dimension of feature space $n=40$ can get a high performance response.

4 Evaluation and Experiments

In this section, we first give the evaluation metric and the test image dataset. Then we give the experiments on real images.

4.1 Evaluation Criterion

We use detection rate and false positive as the performance evaluation metric. They are defined as:

$$d = \frac{\#correct\ matches}{\#possible\ matches} \quad (7)$$

$$f = \frac{\#false\ matches}{\#detection\ points} \quad (8)$$

Two points v_i and v_j are correct match if the error in relative location is less than 2 pixels, which means v_i and v_j should satisfy $Lv_i - H Lv_j < 2$, Where H is the homography between v_i and v_j . Possible matches are the sum of the number of correct matches and false matches.

4.2 Data Set

We use INRIA dataset for experiments. Images can be available on the Internet¹. A significant amount of noise is added which includes blurring, scale, rotation, view point changes, zoom and illumination changes. We generate the detection ratio vs false positive graph by changing the threshold for the proposed algorithm and SIFT algorithm.

¹ <http://www.inrialpes.fr/people/Mikolajczyk/Database>

4.3 Experiments Results

Because authors in [12] demonstrate that SIFT performs best among the common local descriptors. Then in this section, we give the comparing of the proposed algorithm with the standard SIFT algorithm.

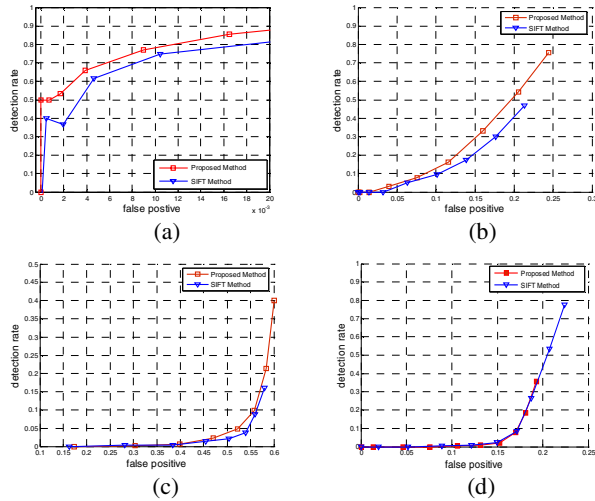


Fig. 2. The performance curves for different image deformations. (a) Scale & rotation changes. (b) View point changes. (c) Illumination changes. (d) Blurring.

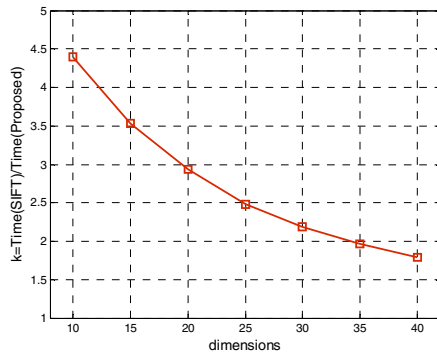


Fig. 3. The running time comparison for different dimensions

In Fig. 2, we give the experimental performances of the proposed algorithm vs SIFT algorithm. The image deformations include scale & rotation, view changes, illumination changes and blurring. We find the proposed method is better than SIFT, especially under scale & rotation changes and view changes. The reason for this phenomenon is that we use EMD to get the feature vectors and train the feature vectors

using ICA. ICA perhaps has the ability to find more instinct features. Researches in [25] have proved our guess. For blurring, the difference between two algorithms is not obvious. The reason for this is that both two algorithms can't find the interest points. Because there are not obvious characters such as corners due to blurring. Anyway, the proposed method is about 2 to 3 times more accurate than SIFT.

In Fig. 3, we compare the running time with SIFT. The range of dimension changes from 10 to 40 with 5 intervals. The rate is computed by $Time(SIFT)/Time(Proposed)$. The proposed method is about 1.8 to 3 times faster than SIFT. For the length of dimension, we find an empirical choice with dimension 40 can get a trade-off between precision and running time.

5 Conclusion and Future Work

In this paper, a new interest point descriptors algorithm is presented. The main contribute of this paper is introducing EMD and ICA to local descriptors. We elaborately design an efficient interest point detector and a compact local descriptors representation. The corresponding experiments are promising.

In the future, we will explore the internal relationship between local descriptors and ICA. Some new applications will be applied.

References

1. T. Tuytelaars and L. Van Gool.: Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 1(59). (2004) 61-85
2. K. Mikolajczyk and C. Schmid.: Indexing based on scale invariant interest points. In *Proceedings of the 8th International Conference on Computer Vision*, Vancouver, Canada. (2001) 525-531
3. J. Sivic and A. Zisserman.: Video Google: A Text Retrieval Approach to Object Matching in Videos. *IEEE International Conference on Computer Vision*. (2003) 1470-1477
4. D. Lowe.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60). (2004) 91-110
5. V. Ferrari, T. Tuytelaars and L. Van Gool.: Simultaneous object recognition and segmentation by image exploration. In *Proceedings of the 8th European Conference on Computer Vision*, Prague, Tcheque Republic. (2004) 40-54
6. R. Deriche and G. Giraudon.: A computational approach for corner and vertex detection. *International Journal of Computer Vision*, 10(2). (1993) 101-124
7. S. Baker, S. Nayar and H. Murase.: Parametric feature detection. *International Journal of Computer Vision*, 27(1). (1998) 27-50
8. P. Kovesi.: Image features from phase congruency. *Videre: A Journal of Computer Vision Research*, MIT press, 1(3). (1999)
9. R. Laganière.: Morphological corner detection. In *Proceedings of International conference on Computer Vision*. (1998) 280-285
10. A. Johnson and M. Hebert.: Object recognition by matching oriented points. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Puerto Rico, USA. (1997) 684-689

11. R. Zabih and J. Woodliff.: Non-parametric local transforms for computing visual correspondence. In Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden. (1994) 151-158
12. K. Mikolajczyk and C. Schmid.: A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10). (2005) 1615-1630
13. S. Belongie, J. Malik and J. Puzicha.: Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4). (2002) 509-522
14. W. Freeman and E. Adelson.: The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9). (1991) 891-906
15. Y. Ke and R. Sukthankar.: PCA-SIFT: A more distinctive representation for local image descriptors. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Washington, USA, 2. (2004) 511-517.
16. J. Koenderink and A. van Doorn.: Representation of local geometry in the visual system. *Biological Cybernetics*, 55. (1987) 367-375
17. S. Lazebnik, C. Schmid and J. Ponce.: Sparse texture representation using affine-invariant neighborhoods. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA. (2003)
18. F. Schaffalitzky and A. Zisserman.: Multi-view matching for unordered image sets. In Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark. (2002)
19. L. Van Gool, T. Moons and D. Ungureanu.: Affine/photometric invariants for planar intensity patterns. In Proceedings of the 4th European Conference on Computer Vision, Cambridge, England. (1996) 642-651
20. C. Schmid, R. Mohr and C. Bauckhage.: Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2). (2000) 151-172
21. C. Harris and M. Stephens.: A combined corner and edge detector. In *Alvey Vision Conference*. (1988) 147-151
22. K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors", *International Journal of Computer Vision*, 1(60). (2004) 63-86
23. N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shin, Q. Zheng, N. C. Yen, C. C. Tung and H. H. Liu.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. Royal Soc. London A*, vol. 454. (1998) 903-995
24. A. Hyvärinen.: Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3). (1999) 626-634
25. J. T. Lindgren and A. Hyvärinen.: Learning high-level independent component of images through a spectral representation. In Proceedings of International conference on Pattern Recognition. (2004) 72-75

Integration of Graph Based Authorization Policies

Chun Ruan¹ and Vijay Varadharajan^{1,2}

¹ School of Computing and Information Technology
University of Western Sydney, Penrith South DC, NSW 1797 Australia
{chun, vijay}@cit.uws.edu.au

² Department of Computing
Macquarie University, North Ryde, NSW 2109 Australia
vijay@ics.mq.edu.au

Abstract. In a distributed environment, authorizations are usually physically stored in several computers connected by a network. Each computer may have its own local policies which could conflict with the others. Therefore how to make a global decision from the local authorization policies is a crucial and practical problem for a distributed system. In this paper, three general integration models based on the degrees of node autonomy are proposed, and different strategies of integrating the local policies into the global policies in each model are systematically discussed. The discussion is based on the weighted authorization graph model that we proposed before.

Keywords: access control, authorization conflict, weighted graph.

1 Introduction

In a distributed environment, authorizations are usually physically stored in several computers connected by a network. Each computer may have its own local policies which could conflict with the others. Therefore how to make a global decision from the local authorization policies is a crucial and practical problem for a distributed system. In this paper, we present different strategies of integrating the local policies into the global policies. The discussion is based on the weighted authorization graph model that we proposed in [5].

There has been extensive research in authentication in distributed computing systems. In contrast, authorization has not received much attention in a distributed context. Woo and Lam [6] proposed a framework for authorization in distributed systems. Their work addresses positive and negative authorizations and provides computable semantics. Abadi et al proposed a modal logic based approach for access control in distributed systems [1]. Their work focuses on how to believe that a principal (subject) is making a request, either on his/her own or on someone else's behalf. However, in a distributed environment composed of connected independent computers which are managed by separate administrative authorities, how to provide a unified global view of authorization is not well

addressed in the research area. This paper is focused on this problem. We study how to integrate the existing local authorization policies into the global one.

On the other hand, the use of graph-based framework to specify and evaluate the authorizations has several advantages. It provides the intuition through visualizing a system's access control policies, the formal basis for the precise semantics of the access control policies, and the great potential to apply existing graph theory results to the domain of authorization. Different graph-based models have been proposed over the years. The take-grant model [4] is among the earliest applications of graph theory to access control. Two special rights take and grant are introduced to transfer the rights from one subject to another. Jaeger and Tidswell have used graphs to model constraints [2]. Koch et al have used graph to specify the administration of RBAC systems [3]. We have presented a weighted graph based model (WGBM) in [5] which allows the grantors to assign a weight to their authorization grants. A conflict resolution method based on the shorter weighted path-take-precedence is proposed, which allows the administrators to control their delegations and grants in a very flexible way.

In this paper, we apply WGBM in a distributed environment. One approach is to keep the global authorization state consistent through creating the global WGBM. Three different ways to build the global WGBM are presented. This method can provide consistent authorization decisions over the network, but the amount of communication involved is often significantly large. Another method is to keep the local authorization states consistent only. In this case, how to solve the conflicts between the local policies becomes crucial. Different strategies to integrate local WGBMs into global decisions are discussed in detail.

The remainder of the paper is organized as follows. Section 2 gives a brief review of WGBM. Section 3 discusses issues about applying WGBM in a distributed environment, while Section 4 presents different strategies to integrate local policies into the global decision. Section 5 concludes the paper.

2 A Brief Review of WGBM

In this section we give a brief review of the weighted graph based model for authorization and delegation [5].

An *Authorization* is of the form $(s, o, t, r, g) : w$, where s, o, t, r, g denote subject, object, grant type, access right, and grantor respectively. w is a non-negative integer which denotes the weight of the authorization. Intuitively, an authorization $(s, o, t, r, g) : w$ states that grantor g has granted subject s the access right r of type t on object o with the weight w . Lesser the weight, the higher priority it denotes. We consider three grant types: *delegatable*(*), *positive*(+) and *negative*(-). * means that the subject has been granted the right to grant r on o as well as the access right r on o . + means that the subject has been granted the access right r on o , while - means that the subject has been denied the access right r on o . An *authorization state*, denoted by $\mathcal{A}$, is the set of all authorizations at a given time.

A weighted digraph is used to represent an authorization state. For every object o and access right r , let $G_{o,r}$ represents all the authorizations on o and r such that for each weighted authorization $(s, o, t, r, g) : w$, there is an arc of type t from g to s in $G_{o,r}$ labelled with w . We use a solid arc to denote $*$, dotted arc to $+$, and an arc with a number of small vertical lines to $-$.

We say that an authorization state $\mathcal{A}$ is *delegation correct*, if a subject s can grant r on o to other subjects only if s is the owner; or s has been granted r of delegatable type $*$ on o . Two authorizations are contradictory if a grantor gives the same subject two different types of authorizations over the same object and access right. That is, $(s, o, t, r, g) : w$ and $(s', o', t', r', g') : w'$ are *contradictory* if $s = s', o = o', r = r', g = g'$, but $t \neq t'$.

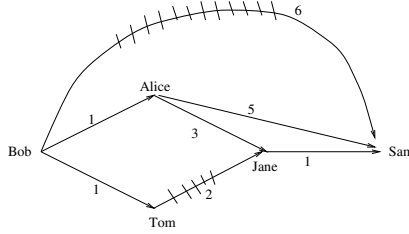
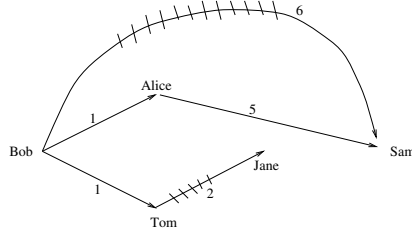
An authorization state $\mathcal{A}$ is *not contradictory* if for any a and a' in $\mathcal{A}$, a and a' are not contradictory. An authorization state is **consistent** if it is delegation correct and not contradictory.

Due to the existence of multiple grantors, conflicts may occur in a consistent state where a subject receives two different types of an access right from different grantors. More formally, two authorizations $(s, o, t, r, g) : w$ and $(s', o', t', r', g') : w'$ are *conflicting* if $s = s', o = o', r = r', t \neq t'$ and $g \neq g'$. Conflicts must be resolved properly in a decentralized authorization model. In our proposed method, we try to find a shortest path from the root to a subject in $G_{o,r}$ which is able to keep the resulting state delegation correct, since lower weight denotes higher priority. A *useful path* is defined for this purpose. A *useful path* to any vertex s in $G_{o,r}$ is a path: $s_o, s_1, \dots, s_k, s$ ($k \geq 0$), where $s_o, s_1, \dots, s_k$ is a delegation path as well as a shortest path to s_k when $k > 0$. To solve the conflicts, only the arcs on useful paths can compete with each other, and the arc in the shortest useful path will override the others. An authorization $(s, o, t, r, g) : w$ is *active* if the corresponding arc (g, s) in $G_{o,r}$ is in a shortest useful path to s (therefore it will not be overridden). Otherwise it is inactive. Please note that, an active path to a vertex s is not necessarily the shortest path from s_o to s , but is definitely the shortest useful path from s_o to s . Let $\mathcal{A}$ be a consistent authorization state, then the subset of all the active authorizations forms its **effective authorization state**, denoted by $Eff(\mathcal{A})$.

There may exist multiple active paths to a vertex s if they have the same shortest length, as is the case of *Sam* in Figure 1. Therefore conflicts may still exist in an effective state. In this case, we say the conflicts are *incomparable*. Incomparable conflicts can be resolved based on the types of authorizations, such as $- > + > *$ which favors security, or the other way around in favor of accessibility. If an authorization state $\mathcal{A}$ is consistent, then the maximal consistent and conflict-free subset of $Eff(\mathcal{A})$ forms a **stable authorization state** of $\mathcal{A}$, denoted as $stable(\mathcal{A})$.

Example 1. Suppose a Patient Bob delegates the Read right on his health file (F) to his two doctors Alice and Tom with the same weight 1. Bob also denies the health researcher Sam to access his record with weight 6. Jane is a health

consultant. Alice further delegates the Read right to Jane with weight 3, but Tom denies Jane to read with weight 2. Both Alice and Jane delegate the Read right to Sam with weight 5 and 1 respectively. Thus we have following authorizations: $(Alice, F, *, R, Bob) : 1$, $(Tom, F, *, R, Bob) : 1$, $(Jane, F, -, R, Tom) : 2$, $(Jane, F, *, R, Alice) : 3$, $(Sam, F, *, R, Jane) : 1$, $(Sam, F, -, R, Bob) : 6$, $(Sam, F, *, R, Alice) : 5$. The corresponding $G_{F,R}$ is shown in Figure 1. The effective authorization state of Figure 1 is shown in Figure 2. There are two stable authorization states of Figure 2. One can be obtained by removing (Bob, Sam) from Figure 2, and the other by removing $(Alice, Sam)$. The former favors accessibility, while the latter favors security.

Fig. 1. $G_{F,R}$ Fig. 2. $\text{Eff}(G_{F,R})$

3 Issues in a Distributed Environment

In the following, we will refer to a computer in a network as a node. We will also omit the authorization weight whenever there is no confusion in the context.

Definition 1. A local authorization state is the set of all authorizations at one node at a given time. A global authorization state is the set of all authorizations at all nodes in a network at a given time.

Remark 1: The global authorization state may not be consistent even though all local authorization states are consistent.

Example 2. Suppose Figure 3 denotes two local authorization states and their corresponding global authorization state for access right R on object F whose

owner is Bob. It is shown that the two local authorization states are both consistent (delegation correct and not contradictory). However the global state is not consistent, since Alice has given Tom two contradictory authorizations.

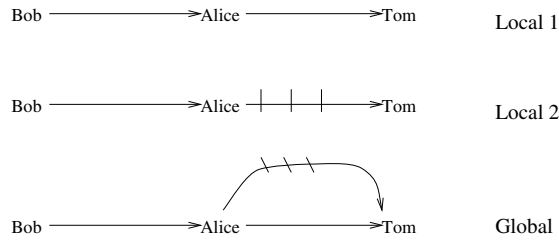


Fig. 3. Consistent Local States But Inconsistent Global State

Remark 2: The global authorization state may be consistent even though some local authorization states are not consistent.

Example 3. In Figure 4, two local authorization states and the related global state for access right R on object F are shown. It is easy to see that the local state of node 2 is not consistent since it is not delegation correct. This is because Alice is neither the owner of F nor the grantee of the delegatable type of R on F , and therefore has no right to grant R on F to Sam. However the global state is consistent, as Alice has been given the right to grant by Bob at node 1.

In general, we have two methods to apply the weighted graph based authorization model in a distributed environment, keeping the global authorization state consistent or only keeping the local authorization state consistent. Both of them have their own advantages and disadvantages in terms of the security, complexity, and cost.

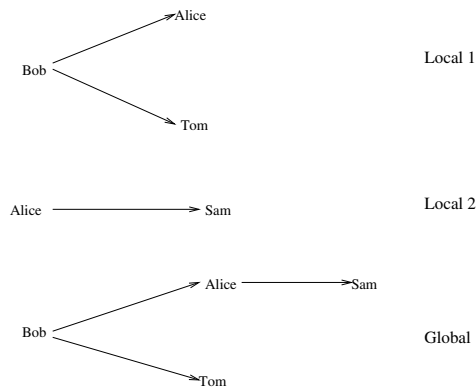


Fig. 4. Inconsistent Local States But Consistent Global State

4 Weighted Authorization Graph Model in a Distributed Environment

In this section, we discuss different approaches to incorporating local authorization policies into the global policy.

4.1 Global Access Control Model

In some distributed processing frameworks, information is logically related but physically distributed to different computers. Physical distribution of data and authorizations are usually transparent to users. That is, from users' point of view, the system behaves like all data and authorizations are stored in the same computer. There is a global access control to process distributed queries, as is the case with (homogeneous) distributed database systems. Thus, the objective in this case is to keep the global authorization state consistent, and evaluate effective and stable states based on it.

Let G denote a graph corresponding to a global authorization state, then G consists of a union of disjoint subgraphs $G_{o,r}$ for all objects o and access rights r . Obviously, G is consistent if and only if $G_{o,r}$ is consistent for each object o and r . We say an authorization $grant(s, o, t, r, g)$ is *local* if it is stored at the node where the object o is stored. Otherwise it is *distributed*. Then, we have the following theorem.

Theorem 1. *If all authorizations are local, and there is a single copy for each object in the network, then the global authorization state is consistent if and only if all local authorization states are consistent.*

Proof. The computer where the object is stored contains all the authorizations on this object, since there is only a single copy of the object in the network and all the authorizations are local. Thus $G_{o,r}$ is entirely located on one node for any o and r . This proves the theorem.

When there exist distributed authorizations or multiple copies of the objects, we need to collect and combine authorizations over the network to evaluate the global authorization state. The possible solutions are discussed below.

- Select one node in the network as the centralised authorization server, and send all authorizations to this node. In this case, the authorization server maintains the global policy and enforces access control for the whole network. All queries need to be sent to the server. This method for example is suitable for the Client-Server network architecture. Fig 5 shows this structure.
- Broadcast each authorization to all nodes in the network. In this case, every node in the network maintains a global state and all queries are processed locally. The advantages of this method include fast query processing and state reliability (if one node fails for some reason, other nodes still keep a complete copy of an authorization state), while the disadvantages are the heavy communication amount and big storage space. For instance, this method is suitable for local area networks where broadcasting is available. Fig 6 shows this structure.

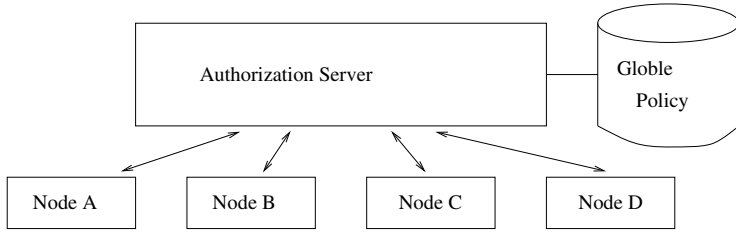


Fig. 5. Single Global Policy

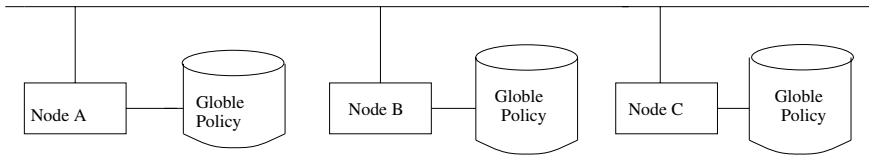


Fig. 6. Broadcasting Authorization Policy

- Send authorizations to the node where the object is stored. When there are multiple copies of objects, we can (1) Send authorizations to all the nodes where the object is stored; (2) Select one copy of the object as the primary copy and send all authorizations to the node where the primary copy is stored. This method is in between the previous two methods. In this case, the global state is partitioned and each node where the object or primary object is stored maintains part of the global state. That is, the node containing the object o (or its primary copy) has a complete $G_{o,r}$ for any r . All queries need to be sent to the nodes where the related objects (or primary objects) reside. Each node may act as an authorization server (for objects that are stored on it) as well as a client (for objects that are not stored on it).
- Divide objects into different classes and select an authorization server for each class. The classes may have intersections, which means that multiple servers may exist for authorizations on a single object. All the authorizations

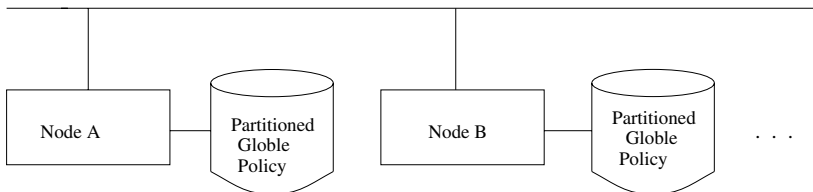


Fig. 7. Partitioned Global Authorization Policy

and queries should be sent to the related class server(s). This method is usually application oriented. We can classify the objects, for example, based on ownership, application or organization. That is, select an authorization server for all the objects owned by a subject, or an application, or an organization. The reason of doing this is that these objects are usually closely related and are thus much more possible to be accessed simultaneously. Fig 7 shows this structure.

4.2 Complete Local Access Control Model

In some network architecture, each node of the network is completely autonomous and does not rely on the global data access control to process distributed queries. Each node can join or leave the system at will without affecting the other nodes, as is the case with federated databases. In this case, we only need to keep the local authorization state consistent and evaluate local effective and stable state based on it. Queries are processed locally based on the local stable state. It is rather flexible. However, since conflicting authorizations may exist at different nodes, the decision to a query may be different when it is submitted at different nodes.

4.3 Mixed Access Control Model

In some networked system, there are both some degree of global control and some degree of autonomy for each node in the network. In this case, each node can maintain its own local authorization state, keeping it consistent and evaluating the effective and stable states based on it. Since conflicting authorizations may still exist at different local states, we need to consider all the local stable states when making a decision to a query. Suppose $S_1, S_2, \dots, S_n$ are n local stable states, we can use the following integration strategies for distributed access control.

(1) Negative take precedence

A request is refused if a related negative authorization exists at any local stable state. In this case, when a node receives a query request, it will check its own local stable state first. If there is a negative authorization about this request, then the request is refused. Otherwise, it needs to send the query to other nodes. If there is a negative authorization in any node, the query is refused. Otherwise, if a positive or delegatable authorization exists at any node then the query is granted. Otherwise, if there is no related authorization, then the query is “undecided”. The “undecided” query will be processed based on the so-called closed or open world assumption. The former will refuse the query while the latter will grant the query.

(2) Positive take precedence

A request is granted if a related positive authorization exists at any local stable state. In this case, when a node receives a query request, it will check its own local stable state first to see if there exists a positive or delegatable authorization about this request. If this is true, then the request is granted. Otherwise, it needs

to send the query to other nodes. If there is a positive or delegatable authorization in any node, the query is granted. Otherwise, if a negative authorization exists at any node, then the query is denied. Otherwise, the query is “undecided”.

(3) All negative required

A request is refused if the related negative authorization exists at all local stable states. In this case, each query will be sent to all the nodes and denied only when a negative authorization about this query exists at all sites. Otherwise, if a positive or delegatable authorization exists at any node then the query is granted. Otherwise, the query is “undecided”.

(4) All positive required

A request is granted if the related positive or delegatable authorization exists at all local stable states. In this case, each query will be sent to all the nodes and granted only when a positive or delegatable authorization about this query exists at all sites. Otherwise, if a negative authorization exists at any node then the query is denied. Otherwise, the query is “undecided”.

(5) Hierarchical structure of nodes

In this case, we consider the priorities of nodes. That is, nodes are organised into a hierarchy (partial order) where lower level means higher priority. Thus, for the conflicting authorizations from nodes at different levels, the lower level one will override the higher level one. For the conflicting authorizations from nodes at the same level, we can still use the methods stated above to resolve them. Therefore, when a node receives a query, it only needs to send it to the nodes which are lower or equal to it to make a decision. Fig 8 shows this structure.

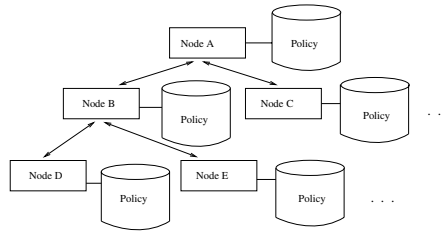


Fig. 8. The Hierarchical Authorization Policy

5 Conclusion

In this paper we apply the weighted graph based authorization model in a distributed environment. Three general models based on the degrees of node autonomy are proposed namely global access control model, complete local access control model and mixed access control model. Different strategies of integrating the local policies into the global policies under each model are discussed in detail. For future work, we plan to investigate approaches to integrating logic based local policies.

References

1. M. Abadi, M. Burrows, B. Lampson, G. Plotkin, A calculus for access control in distributed systems. *ACM Trans. on programming languages and systems*, 15(4): 706-734, 1993.
2. T. Jaeger and J. E. Tidswell. Practical safety in flexible access control models. *ACM Trans. on Info. and System Security*, 4(2):158-190, 2001.
3. M. Koch, L. V. Mancini and F. Parisi-Presicce. Administrative Scope in the Graph-Based Framework. *Proceedings of the ninth ACM Symposium on Access control Models and Technologies*, pp 97-104, 2004.
4. R.J. Lipton and L. Snyder. A Linear Time Algorithm for Deciding Subject Security. *Journal of the ACM*, 24(3):455-464, 1977.
5. C. Ruan and V. Varadharajan, A weighted graph approach to authorization delegation and conflict resolution. in Proceedings of the 9th Australasian Conference on Information Security and Privacy, pp 402-413, 2004.
6. T. Woo and S. Lam, Authorization in distributed systems: a formal approach. *Proceedings of IEEE on Research in Security and Privacy*, pp33-50, 1992.

Supporting Visual Exploration of Discovered Association Rules Through Multi-Dimensional Scaling

Margherita Berardi¹, Annalisa Appice¹, Corrado Loglisci¹, and Pietro Leo²

¹ Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70126 Bari - Italy

² IBM GBS - Innovation Center - Bari

Via Tridente, 42/14 - 70125 Bari

{berardi, appice, loglisci}@di.uniba.it,
pietro_leo@it.ibm.com

Abstract. Association rules are typically evaluated in terms of support and confidence measures, which ensure that discovered rules have enough positive evidence. However, in real-world applications, even considering only those rules with high confidence and support it is not true that all of them are interesting. It may happen that the presentation of all discovered rules can discourage users from interpreting them in order to find nuggets of knowledge. Association rules interpretation can benefit from discovering group of “similar” rules, where (dis)similarity is estimated on the basis of syntactic or semantic characteristics. In this paper, we resort to the multi-dimensional scaling to support a visual exploration of association rules by means of bi-dimensional scatter-plots. An application in the domain of biomedical literature is reported. Results show that the use of this visualization technique is beneficial.

1 Introduction

Association rules are a class of regularities introduced by [1], which are expressed in the form $A \Rightarrow C$, where both the antecedent A and the consequent C are sets of items such that $A \cap C = \emptyset$. The meaning of such rules is quite intuitive: Given a dataset of transactions D , where each transaction $T \in D$ is a set of items, $A \Rightarrow C$ expresses that whenever a transaction T contains A than T probably contains C . The conjunction $A \cup C$ is called pattern. Two parameters are usually reported for association rules, namely the support that estimates the probability $p(A \subseteq T \wedge C \subseteq T)$, and the confidence that estimates the probability $p(C \subseteq T | A \subseteq T)$. The goal of association rule mining is to discover all the rules with support and confidence exceeding user specified thresholds, henceforth called minsup and minconf, respectively. Although support and confidence measures ensure that the discovered association rules have enough positive evidence, the usually high number of rules may represent an obstacle to achieve more meaningful and rigorous data analysis in association rules discovery tasks.

One important problem is that users may not have the time and knowledge required to adequately understand the dynamics and operation of association rule discovery process. This problem has been reflected in a lack of sound practices for exploring association rules in order to assess their statistical significance. Indeed, relatively less attention has been given to exploratory, visualisation-driven techniques to support the identification of key patterns from data. This task refers to the recognition of key groups of patterns, outliers and features based on computationally-inexpensive, user-friendly and robust analysis. Its outcomes may offer guidance to gain a better insight into the high-level structure and associations found in the data. In this context, a visualization-based exploratory approach may generate useful alternative views for supporting a more intelligent and meaningful application of association rules.

An important data visualization approach comprises the application of non-linear mapping techniques based on the idea of transforming an original, n -dimensional input space into a reduced, m -dimensional one, where $m < n$. These methods, known as non-linear projection methods or multi-dimensional scaling methods [4], aim to represent higher dimensional data (e.g., association rules) by a more meaningful lower dimensional structure and preserve global properties, such as distances, in the transformed, m -dimensional space [7,8,3].

In this paper, we propose to use the Sammon's multi-dimensional scaling [8] to map association rules into points of a 2-dimensional Euclidean area. The starting point is a dissimilarity matrix that contains the pairwise dissimilarities estimated on the basis of either syntactic or semantic characteristics of association rules under projection. Sammon's mapping is performed in such a way that the Euclidean distances between points preserve these dissimilarities as well as possible. Finally, points are visualized within a bi-dimensional scatter-plot.

The paper is organized as follows. In the next Section, we present several (dis)similarity measures that resort to a set interpretation of association rules and compare them on the basis of their syntactic or semantic characteristics. In Section 3, we present the association rule visualization in a bi-dimensional scatter-plot. An application in the domain of biomedical literature is reported in Section 4, and conclusion and future works are drawn in Section 5.

2 (Dis)similarity Measures Between Association Rules

A similarity measure s on a set of objects (e.g., association rules) E is a real valued function $s : E \times E \rightarrow \mathbb{R}$ such that $s_a^* = s(a, a) \geq s(a, b) = s(b, a) \geq 0$ for all $a, b \in E$. Differently, a dissimilarity measure d is a real valued function $d : E \times E \rightarrow \mathbb{R}$, such that $d_a^* = d(a, a) \leq d(a, b) = d(b, a) < \infty$ for all $a, b \in E$. Generally, $s_a^* = s^*$ and $d_a^* = d^*$ for each $a \in E$, and more specifically, $s^* = s(a, a) = 1$, while $d^* = d(a, a) = 0$ [2]. A similarity measure s can be transformed into a dissimilarity one d by preserving properties defined with the induced quasi-ordering. This transformation is made possible by imposing $d = f(s)$ and $s = y(d)$, respectively, where $f(\bullet)$ and $y(\bullet)$ are strictly decreasing functions with boundary conditions. A commonly used transformation is:

$$d = \max(s) - s, \quad (1)$$

where $\max(s)$ is the maximum observable value of s . Insofar, we restrict our discussion to similarity measures only and then derive the dissimilarity judgement.

In this work, we consider E as a set of association rules $R_1 : A_1 \Rightarrow C_1, \dots, R_n : A_n \Rightarrow C_n$ discovered from a non empty dataset $D(X)$ with $X = \{X_1, \dots, X_m\}$. Each association rule $A_i \Rightarrow C_i \in E$ is composed of a syntactic part and a semantic one [9]. The former describes a relation involving sets of attributes from D , that is, an association between $A_i \subset \{X_1, \dots, X_n\}$ and $C_i \subset \{X_1, \dots, X_n\}$ ($A_i \cap C_i = \emptyset$). The latter includes the probabilistic indexes (*support* ($A_i \Rightarrow C_i$) $\geq \text{minsup}$ and *confidence* ($A_i \Rightarrow C_i$) $\geq \text{minconf}$) with relative supporting sets in D . Hence, the (dis)similarity judgment between R_i and R_j is based on the comparison of the syntactic (or semantic) parts of the association rules.

2.1 Syntactic (dis)similarity

The judgment of the syntactic (dis)similarity between the association rules R_1 and R_2 is performed by comparing R_1 and R_2 in order to identify the items that equally (or differently) occur in antecedent (A_1 and A_2) and consequent (C_1 and C_2) parts of both association rules.

Example 1. Let us consider the association rules:

$R_1: \text{AlzheimerDisease}, \text{MemoryDisorders} \Rightarrow \text{Depression}, \text{Cholesterol}$
(support = 0.8, confidence = 0.72)

$R_2: \text{AlzheimerDisease}, \text{MemoryDisorders} \Rightarrow \text{Depression}, \text{Brain}$
(support = 0.95, confidence = 0.8)

which describe some relation among the terms that co-occur within a set of bio-medical scientific abstracts. R_1 states that the terms “Depression” and “Cholesterol” are observed in the same abstracts that also include the terms “Alzheimer Disease” and “Memory Disorders”, while R_2 states that the terms “Depression” and “Brain” are observed in the same abstracts that also include the terms “Alzheimer Disease” and “Memory Disorders”. By ignoring the values of support and confidence, the difference between R_1 and R_2 are the terms “Cholesterol” and “Brain” that occur differently in the consequent parts of these rules.

Focusing on the similarity judgement, the syntactic similarity between R_1 and R_2 is estimated by comparing the antecedent parts (A_1, A_2) and the consequent parts (C_1, C_2) separately and then returning the average value as follows:

$$s(R_1, R_2) = \frac{s(A_1, A_2) + s(C_1, C_2)}{2} \quad (2)$$

This is quite different from the work of [9], where the syntactic similarity between R_1 and R_2 is estimated by comparing only the antecedent parts of the association rules and imposing that R_1 and R_2 have exactly the same consequent.

	P_1	P_2	
P_1	a	b	a+b
P_2	c	d	c+d
	a+c	b+d	a+b+c+d = #X=m

Fig. 1. The contingency table of the similarity comparison

In alternative, the similarity between R_1 and R_2 is performed by directly comparing the entire patterns ($A_1 \cup C_1$ and $A_2 \cup C_2$) underlying R_1 and R_2 , that is:

$$s(R_1, R_2) = s(A_1 \cup C_1, A_2 \cup C_2) \quad (3)$$

In both cases, $s(R_1, R_2)$ is computed by resorting to a set-theoretic interpretation [10] of association rules. In the former case, each association rule R_i is described by maintaining separate the antecedent set (A_i) from the consequent set (C_i), while in the latter case, R_i is described by means of the entire pattern set ($A_i \cup C_i$). In particular, let $P_1 = \{V_1, \dots, V_h\}$ ($h \geq 1$) and $P_2 = \{W_1, \dots, W_k\}$ ($k \geq 1$) be two set of items on X ($P_1 \subseteq X$ and $P_2 \subseteq X$), $s(P_1, P_2)$ is estimated by resorting to the cardinalities:

$$\begin{aligned} a &= \#(P_1 \cap P_2), \\ b &= \#(P_1 - P_2), \\ c &= \#(P_2 - P_1), \\ d &= \#(X - (P_1 \cup P_2)), \end{aligned}$$

and building the contingency table representing the quantitative information about the syntactic aspects of P_1 and P_2 . Starting with the contingency table in Figure 1, several similarity measures are defined in statistics. In this paper, we consider the following similarity measures [9]:

1. Jaccard's coefficient: $jaccard(P_1, P_2) = \frac{a}{(a+b+c)}$,
2. Kulczynski's coefficient: $kulczynski(P_1, P_2) = \frac{1}{2}(\frac{a}{a+b}) + \frac{a}{a+c}$,
3. Ochiai's coefficient: $ochiai(P_1, P_2) = \frac{a}{\sqrt{(a+b)(a+c)}}$,
4. Simpson's coefficient: $simpson(P_1, P_2) = \frac{a}{\min\{(a+b), (a+c)\}}$.

which satisfy the property of symmetry and return a similarity value within the interval $[0, 1]$. This makes possible to derive a dissimilarity judgement from a similarity one as follows:

$$d(R_1, R_2) = 1 - s(R_1, R_2), \quad (4)$$

where $1 = \max(s)$.

2.2 Semantic (dis)similarity

The semantic (dis)similarity between the association rules R_1 and R_2 is computed by resorting to a *semantic-based* set-interpretation of association rules to compare the meaning of rules from the viewpoint of the supporting sets.

Definition 1. Let P be a set of items, the supporting set of P on D , denoted by $set(P, D)$, is the set of transactions $T \in D$ such that T contains P .

Example 2. Let us consider the set of items $P = \{Alzheimer\ Disease, Brain\}$ and the dataset D defined on $X = \{Alzheimer\ Disease, Brain, Cholesterol, Depression, Memory\ Disorders\}$, that is:

Nr	Alzheimer Disease	Brain	Cholesterol	Depression	Memory Disorders
1	true	true	false	false	true
2	true	false	false	false	false
3	true	true	true	true	true

the supporting set of P on D is $set(P, D) = \{1, 3\}$.

According to the definition of supporting set, an association rule $R_i : A_i \Rightarrow C_i$ can be semantically described with respect to D in terms of:

1. the pair of supporting sets $\langle set(A_i, D), set(C_i, D) \rangle$ associated with the antecedent part and the consequent part of R_i , respectively,
2. or, the supporting set $set(A_i \cup C_i, D)$ associated with the entire pattern $A_i \cup C_i$ of R_i .

In the former case, the semantic similarity between R_1 and R_2 is computed by comparing separately the supporting sets of the antecedent parts (i.e., $s(set(A_1, D), set(A_2, D))$) and the consequent parts (i.e., $s(set(C_1, D), set(C_2, D))$) and returning the average similarity value as follows:

$$s(R_1, R_2) = \frac{s(set(A_1, D), set(A_2, D)) + s(set(C_1, D), set(C_2, D))}{2} \quad (5)$$

Conversely, in the latter case, the supporting sets associated with the entire patterns underlying R_1 and R_2 are compared as follows:

$$s(R_1, R_2) = s(set(A_1 \cup C_1, D), set(A_2 \cup C_2, D)) \quad (6)$$

In both cases, the supporting sets of P_1 and P_2 are compared on the basis of Jaccard's coefficient, Kulczynsky's coefficient, Ochiai's coefficient or Simpson's coefficient, while the table of contingency is built by imposing:

$$\begin{aligned} a &= \#(set(P_1, D) \cap set(P_2, D)), \\ b &= \#(set(P_1, D) - set(P_2, D)), \\ c &= \#(set(P_2, D) - set(P_1, D)), \\ d &= \#(D - (set(P_2, D) \cup set(P_1, D))). \end{aligned}$$

The semantic dissimilarity value is then obtained by complementing the semantic similarity value with respect to 1.

3 Multi-Dimensional Scaling Based Visualization

The multi-dimensional scaling (MDS) refers to a class of algorithms for the exploratory data analysis, which visualize the dissimilarities between objects by means of distances between points of a low-dimensional (usually, two dimensional) Euclidean space [6]. This allows to provide data miners with exploration tools that immediately reveal the inherent structure of data and allow to visually discover possible clusters, correlations or underlying distributions.

This idea is profitably applied to the case that the objects to be explored are a set of association rules $E = \{R_1, \dots, R_n\}$ and the MDS is used to map each association rule $R_i \in E$ into a point $B_i(x_i, y_i) \in \mathbb{R}^2$ of a 2-dimensional Euclidean space. This mapping is performed according to the Sammon's MDS [8] by preserving the inherent structure of rules, that is, the (syntactic or semantic) dissimilarities between association rules under projection.

The starting point is a matrix $DS(n \times n)$ consisting of the pairwise dissimilarities $ds_{ij} = d(R_i, R_j)$ computed by comparing R_i and R_j for each $i, j = 1 \dots n$. Sammon's mapping is then performed in such a way that the Euclidean distances $euclidean(B_i, B_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ approximate the original dissimilarity values ds_{ij} in DS as well as possible. This is obtained by solving a minimization problem and optimizing a cost (stress) function that describes how well the pairwise distances (dissimilarities vs. Euclidean distances) in E are preserved. The stress function is given by the following formula:

$$stress = \frac{\sum_{i=1 \dots n-1} \sum_{j=i+1 \dots n} \frac{(euclidean(B_i, B_j) - ds_{ij})^2}{euclidean(B_i, B_j)}}{\sum_{i=1 \dots n-1} \sum_{j=i+1 \dots n} euclidean(B_i, B_j)} \quad (7)$$

and yields in fact a badness-of-fit measure for the entire representation. The stress range is $[0, 1]$ with 0 indicating a lossless mapping.

The minimum Sammon's stress is searched by applying an iterative hill-climbing where points mapping association rules are firstly assigned with random coordinates in $\mathbb{R}^2$ and iteratively moved by small distance in direction of locally decreasing stress function according to the gradient-descent procedure. Anyway, this may lead to reach only a "local" minimum in the stress surface. To overcome this limitation, a significant number of experiments with different random initializations should be performed to obtain appropriate results.

Points returned by the Sammon's MDS are visualized in a bi-dimensional scatter-plot area. This is different from the exploration approach, well-known to data miners, that consists in applying clustering methods [5] to identify objects sharing a similar structure in E . Clustering may be inefficient and typically pose the problem of rendering the discovered clusters in a human-interpretable form, while this rendering is free obtained within a scatter-plot visualization.

Example 3. Let us consider the set of association rules $E = \{R_1, R_2, R_3, R_4\}$ such that $R_1 : a \Rightarrow b, c, d$; $R_2 : b, c \Rightarrow a, d$; $R_3 : a, b, e \Rightarrow f$ and $R_4 : e, f, g \Rightarrow a, b$. Patterns associated to each pair of association rules $(R_i, R_j) \in E \times E$ are syn-

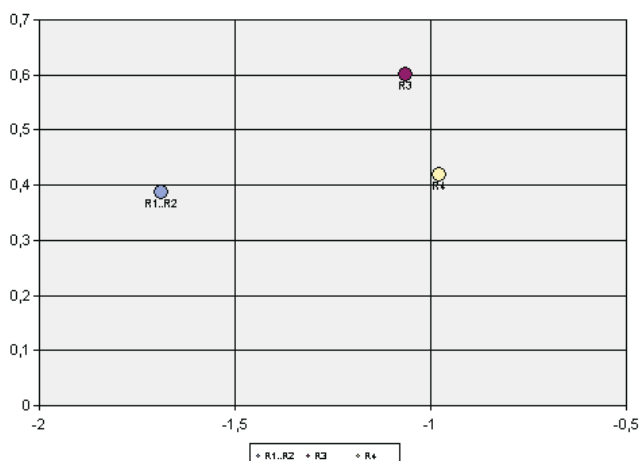


Fig. 2. The Sammon's MDS of a set of association rules and the visualization of output points in a bi-dimensional scatter-plot area

tactically compared by the Jaccard's coefficient and the following dissimilarity matrix D is built:

$$D = \begin{pmatrix} 0 & 0 & 0.66 & 0.71 \\ 0 & 0 & 0.66 & 0.71 \\ 0.66 & 0.66 & 0 & 0.2 \\ 0.71 & 0.71 & 0.2 & 0 \end{pmatrix}.$$

By using the Sammon's MDS, association rules are mapped into points of a bi-dimensional scatter-plot area (see Figure 2) that visually reveals the syntactic similarity between the association rules R_3 and R_4 in addition to the syntactic identity between R_1 and R_2 .

4 Application

In this section, we present an application of the proposed MDS-based visualization technique to support the exploration of a set of association rules mined from the biomedical literature. The data set is composed by abstracts of scientific publications that are collected by querying PubMed, (<http://www.ncbi.nlm.nih.gov/entrez/>), the search engine interfacing Medline, that is, the main resource of biomedical research literature. Each query generates a different data set that is stored in a single table of a relational database, where each transaction is associated with an individual abstract. The boolean representation is adopted to represent the occurrence of a term (i.e., biomedical named entity) in an abstract. More precisely, we consider only the most frequent terms (about 50) with respect to the set of retrieved abstracts. In this study, the Medline segment corresponding to the query “Alzheimer Drug Treatment Response” is

considered, that returns 499 abstracts. 187 association rules are mined by using Apriori algorithm [1] and tuning threshold values for minimum support and minimum confidence to 0.2 and 0.7, respectively. Dissimilarity matrices are built according to both syntactic and semantic (dis)similarity measures presented in Section 2. In particular, rule (dis)similarity is computed by directly comparing the underlying patterns or returning the average (dis)similarity value obtained by comparing both antecedent and consequent parts of the rules. In both cases, the comparison can be performed according to the Jaccard’s coefficient, Kulczynski’s coefficient, Ochiai’s coefficient or Simpson’s coefficient. For each dissimilarity matrix, a bi-dimensional mapping is produced and rendered through a scatter-plot. The MDS technique is repeated until the minimization of the mapping error (namely *stressfunction* before defined) leads to results visually satisfying user expectations.

Due to space limitation, we report only scatter-plots for the Sammmon’s MDS obtained by considering pairwise dissimilarities semantically estimated on both antecedent and consequent parts of rules (see Figures 3, 4 and 5). For each plot, the mapping error and the number of iterations are reported. Results for the

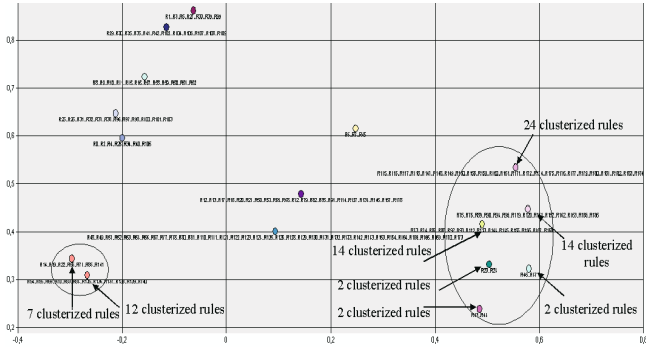


Fig. 3. The visualization of a set of association rules semantically compared by the Jaccard’s coefficient and using average values of antecedent and consequent parts. Mapping Error: 0.01417; Number of Iterations: 288.

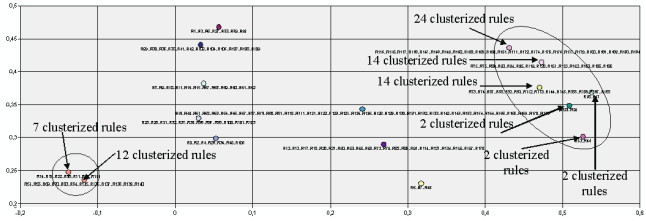


Fig. 4. The visualization of a set of association rules semantically compared by the Kulczynski’s coefficient and using average values of antecedent and consequent parts. Mapping Error: 0.013708; Number of Iterations: 716.

Ochiai's coefficient are not reported since the output scatter-plot is very similar to the one produced with Kulczynsky's coefficient. We observe that several association rules are grouped together into a single point of the bi-dimensional area by revealing semantic identity among rules belonging to the same point. Some clusters of points (manually underlined by circles reported in figures) can also be identified. They reveal a certain degree of similarity among groups of rules. It is interesting to observe that some similarities are in some way disclosed independently on the particular coefficient adopted to compute (dis)similarity measures.

By comparing scatter-plots obtained for syntactic measures, we observe that semantic measures have an higher potential to discover association rules similarities. In particular, when measures are computed on patterns, a total of 6 different points are enough to represent all the 187 rules and no point containing a single rule results. When antecedent and consequent parts are separately considered, the ability of grouping similar association rules is also more evident than in the case of syntactic estimations. On the other hand, MDS on syntactic measures may reveal unexpected cluster of rules with different support values.

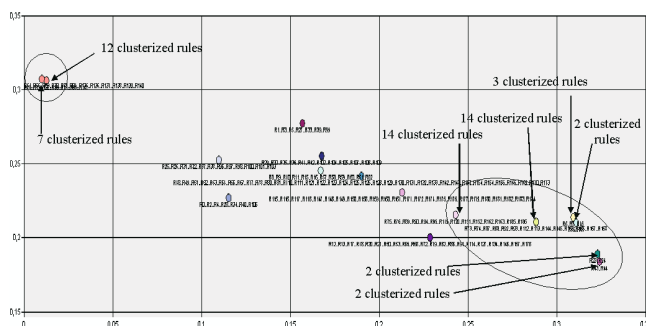


Fig. 5. The visualization of a set of association rules semantically compared by the Simpson's coefficient and using average values of antecedent and consequent parts. Mapping Error: 0.17618; Number of Iterations: 743.

This exploration tool allows data miners to immediately reveal the inherent structure of data. By visually discovering clusters and distributions of interest, the data miner may conduct a more in deep analysis on portions of data. For instance, interesting rules can be selected in a local way as representative points of clusters.

5 Conclusions

In this paper, we propose a multi-dimensional scaling technique to support a visual exploration of association rules by means of bi-dimensional scatter-plots. This allows to provide data miners with exploration tools that immediately reveal

the inherent structure of data and allow to visually discover possible clusters, correlations or underlying distributions. In particular, multi-dimensional scaling is performed according to the Sammon's mapping that allow to represent each association rule as a point in a bi-dimensional Euclidean space and preserve the inherent structure of association rules, that is, the (syntactic or semantic) pairwise dissimilarities.

An application in the domain of biomedical literature is reported. Results show that the association rule visualization in a scatter-plot area is beneficial to explore the huge amount of association rules and discover nuggets of knowledge in form of clusters of rules sharing a similar (syntactic or semantic) structure.

Acknowledgment

This work has been funded by the COFIN 2005 "Metodi di data mining spaziali per sistemi di supporto alle decisioni ambientali".

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Databases*, 1994.
2. V. Batagelj and M. Bren. Comparing resemblance measures. *Journal of Classification*, 12:73–90, 1995.
3. A. Buja and D. F. Swayne. Visualization methodology for multidimensional scaling. *Journal of Classification*, 19:7–43, 2002.
4. T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman and Hall, 1994.
5. B. S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. Edward Arnold, 2001.
6. H. Klock and J. M. Buhmann. Multidimensional scaling by deterministic annealing. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 245–260, 1997.
7. J. Kruskal. Non-metric multidimensional scaling: a numerical method. *Psychometrika*, 29:115–129, 1964.
8. J. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18:401–409, 1969.
9. S. Tsumoto and S. Hirano. Visualization of similarities and dissimilarities in rules using multidimensional scaling. In *Proceedings of the 15th International Symposium on Foundations of Intelligent Systems, ISMIS 2005*, volume 3488 of *Lecture Notes in Computer Science*, pages 38–46. Springer, 2005.
10. Y. Y. Yao and N. Zhong. An analysis of quantitative measures associated with rules. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 479–488, 1999.

Evaluating Learning Algorithms for a Rule Evaluation Support Method Based on Objective Rule Evaluation Indices

Hidenao Abe¹, Shusaku Tsumoto¹, Miho Ohsaki², and Takahira Yamaguchi³

¹ Department of Medical Informatics, Shimane University, School of Medicine
89-1 Enya-cho, Izumo, Shimane 693-8501, Japan
abe@med.shimane-u.ac.jp, tsumoto@computer.org

² Faculty of Engineering, Doshisha University
1-3 Tataramiyakodani, Kyo-Tanabe, Kyoto 610-0321, Japan
mohsaki@mail.doshisha.ac.jp

³ Faculty of Science and Technology, Keio University
3-14-1 Hiyoshi, Kohoku Yokohama, Kanagawa 223-8522, Japan
yamaguti@ae.keio.ac.jp

Abstract. In this paper, we present an evaluation of learning algorithms of a novel rule evaluation support method for post-processing of mined results with rule evaluation models based on objective indices. Post-processing of mined results is one of the key processes in a data mining process. However, it is difficult for human experts to completely evaluate several thousands of rules from a large dataset with noises. To reduce the costs in such rule evaluation task, we have developed the rule evaluation support method with rule evaluation models which learn from a dataset. This dataset comprises objective indices for mined classification rules and evaluations by a human expert for each rule. To evaluate performances of learning algorithms for constructing the rule evaluation models, we have done a case study on the meningitis data mining as an actual problem. Furthermore, we have also evaluated our method with five rule sets obtained from five UCI datasets.

1 Introduction

In recent years, enormous amounts of data are stored on information systems in natural science, social science, and business domains. People have been able to obtain valuable knowledge due to the development of information technology. Besides, data mining techniques combine different kinds of technologies such as database technologies, statistical methods, and machine learning methods. Then, data mining has been well-known for utilizing data stored on database systems. In particular, if-then rules, which are produced by rule induction algorithms, are considered as one of the highly usable and readable outputs of data mining. However, to large datasets with hundreds of attributes including noise, the process often obtains many thousands of rules. From such a large rule set, it is difficult for human experts to find out valuable knowledge which are rarely included in the rule set.

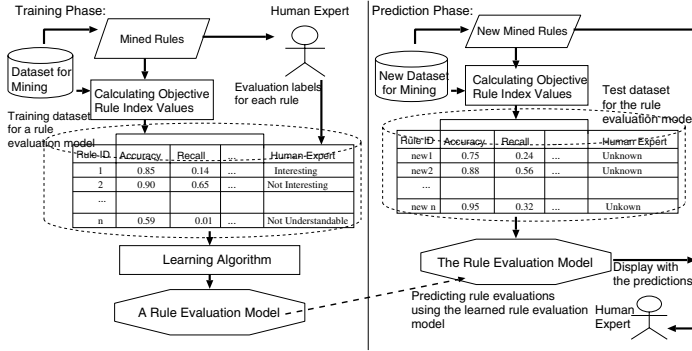


Fig. 1. Overview of the construction method of rule evaluation models

To support such a rule selection, many efforts have done using objective rule evaluation indices such as recall, precision, and other interestingness measurements [11,21,23] (Hereafter, we refer to these indices as “objective indices”). Further, it is difficult to estimate the criterion of a human expert using a single objective rule evaluation index; this is because his/her subjective criterion such as interestingness and importance for his/her purpose is influenced by the amount of his/her knowledge and/or the passage of time.

With regard to the above mentioned issues, we have developed an adaptive rule evaluation support method for human experts with rule evaluation models. This method predicts the experts’ criteria based on objective indices by re-using the results of the evaluations by human experts. In Section 2, we describe the rule evaluation model construction method based on objective indices. Then, we present a performance comparison of learning algorithms for constructing rule evaluation models in Section 3. With the results of the comparison, we discuss the availability of our rule evaluation model construction approach.

2 Rule Evaluation Support with Rule Evaluation Model Based on Objective Indices

We considered the process of modeling rule evaluations of human experts as the process to clarify the relationships between the human evaluations and features of inputted if-then rules. Based on this consideration, we decided that the rule evaluation model construction process can be implemented as a learning task. Fig. 1 shows the rule evaluation model construction process based on the re-use of human evaluations and objective indices for each mined rule.

In the training phase, the attributes of a meta-level training data set are obtained by objective indices such as recall, precision and other rule evaluation values. The human evaluations for each rule are combined as classes of each instance. To obtain this data set, a human expert has to evaluate the whole or a part of the input rules at least once. After obtaining the training data

set, its rule evaluation model is constructed by using a learning algorithm. At the prediction phase, a human expert receives predictions for new rules based on their objective index values. Since rule evaluation models are used for predictions, we need to choose a learning algorithm with high accuracy similar to the current classification problems.

3 Performance Comparisons of Learning Algorithms for Rule Model Construction

To predict human evaluation labels of a new rule based on objective indices more accurately, we have to construct a rule evaluation model with a higher predictive accuracy.

In this section, we first present the result of an empirical evaluation with the dataset obtained from the result of a meningitis data mining [9]. Then, to confirm the performance of our approach on the other datasets, we evaluated five algorithms on five rule sets obtained from five UCI benchmark datasets [10]. Based on the experimental results, we discuss the followings: accuracy of rule evaluation models, analysis of learning curves of the learning algorithms, and contents of the learned rule evaluation models.

For evaluating the accuracy of the rule evaluation models, we have compared predictive accuracies on the entire dataset and Leave-One-Out validation. The accuracy of a validation dataset D is calculated with correctly predicted instances $Correct(D)$ as $Acc(D) = (Correct(D)/|D|) \times 100$, where $|D|$ is the size of the dataset. The recalls of class i on a validation dataset are calculated using correctly predicted instances about the class $Correct(D_i)$ as $Recall(D_i) = (Correct(D_i)/|D_i|) \times 100$, where $|D_i|$ is the size of instances of class i . Further, the precision of class i is calculated using the size of instances which are predicted i as $Precision(D_i) = (Correct(D_i)/Predicted(D_i)) \times 100$.

With regard to the learning curves, we obtained curves of accuracies of learning algorithms on the entire training dataset to evaluate whether each learning algorithm can perform in the early stage of rule evaluation process. Accuracies of randomly sub-sampled training datasets are averaged with 10 trials on each percentage of the subset.

By observing the elements of the rule evaluation models on the meningitis data mining result, we consider the characteristics of the objective indices which are used in these rule evaluation models.

In order to construct a dataset to learn a rule evaluation model, the values of the objective indices have been calculated for each rule by considering 39 objective indices as shown in Table 1. Thus, each dataset for each rule set has the same number of instances as the rule set. Each instance has 40 attributes including those of the class.

We applied five learning algorithms to these datasets to compare their performances as a rule evaluation model construction method. We used the following learning algorithms from Weka [22]: C4.5 decision tree learner[18] called J4.8,

Table 1. Objective rule evaluation indices for classification rules used in this research. **P**: Probability of the antecedent and/or consequent of a rule. **S**: Statistical variable based on P. **I**: Information of the antecedent and/or consequent of a rule. **N**: Number of instances included in the antecedent and/or consequent of a rule. **D**: Distance of a rule from the others based on rule attributes.

Theory	Index Name (Abbreviation)	Reference Number of Literature
P	Coverage (Coverage), Prevalence (Prevalence) Precision (Precision), Recall (Recall) Support (Support), Specificity (Specificity) Accuracy (Accuracy), Lift (Lift) Leverage (Leverage), Added Value (Added Value)[21] Klöggen's Interestingness (KI)[14], Relative Risk (RR)[1] Brin's Interest (BI)[2], Brin's Conviction (BC)[2] Certainty Factor (CF)[21], Jaccard Coefficient (Jaccard)[21] F-Measure (F-M)[19], Odds Ratio (OR)[21] Yule's Q (YuleQ)[21], Yule's Y (YuleY)[21] Kappa (Kappa)[21], Collective Strength (CST)[21] Gray and Orłowska's Interestingness weighting Dependency (GOI)[7] Gini Gain (Gini)[21], Credibility (Credibility)[8]	
S	χ^2 Measure for One Quadrant (χ^2 - M1)[6] χ^2 Measure for Four Quadrant (χ^2 - M4)[6]	
I	J-Measure (J-M)[20], K-Measure (K-M)[15] Mutual Information (MI)[21] Yao and Liu's Interestingness 1 based on one-way support (YLI1)[23] Yao and Liu's Interestingness 2 based on two-way support (YLI2)[23] Yao and Zhong's Interestingness (YZI)[23]	
N	Cosine Similarity (CSI)[21], Laplace Correction (LC)[21] ϕ Coefficient (ϕ)[21], Piatetsky-Shapiro's Interestingness (PSI)[16]	
D	Gago and Bento's Interestingness (GBI)[5] Peculiarity (Peculiarity)[24]	

neural network learner with back propagation (BPNN)[12], support vector machines (SVM)¹[17], classification via linear regressions (CLR)²[3], and OneR [13].

3.1 Constructing Rule Evaluation Models for an Actual Datamining Result

In this case study, we have considered 244 rules, which are mined from six datasets about six types of diagnostic problems as shown in Table 2. In these datasets, appearances of meningitis patients were considered as attributes and the diagnosis of each patient as a class. Each rule set was mined with its proper rule induction algorithm composed by a constructive meta-learning system called CAMLET [9]. For each rule, we labeled three evaluations (I: Interesting, NI: Not-Interesting, NU: Not-Understandable) according to evaluation comments provided by a medical expert.

Comparison of Classification Performances. In this section, we present the result of accuracy comparison over the entire dataset, recall of each class label, and precisions them. Since Leave-One-Out holds just one test instance and the

¹ A polynomial kernel function was used.

² We set up the elimination of collinear attributes and the model selection with greedy search based on the Akaike information metric.

Table 2. Description of the meningitis datasets and the results of datamining

Dataset	#Attributes	#Class	#Mined rules	#'I' rules	#'NI' rules	#'NU' rules
Diag	29	6	53	15	38	0
C_Course	40	12	22	3	18	1
Culture+diag	31	12	57	7	48	2
Diag2	29	2	35	8	27	0
Course	40	2	53	12	38	3
Cult_find	29	2	24	3	18	3
TOTAL	—	—	244	48	187	9

Table 3. Accuracies (%), Recalls (%), and Precisions (%) of the five learning algorithms

	Over the entire training dataset								Leave-One-Out							
	Acc.	Recall			Precision			Acc.	Recall			Precision			Acc.	NU
		I	NI	NU	I	NI	NU		I	NI	NU	I	NI	NU		
J4.8	85.7	41.7	97.9	66.7	80.0	86.3	85.7	79.1	29.2	95.7	0.0	63.6	82.5	0.0		
BPNN	86.9	81.3	89.8	55.6	65.0	94.9	71.4	77.5	39.6	90.9	0.0	50.0	85.9	0.0		
SVM	81.6	35.4	97.3	0.0	68.0	83.5	0.0	81.6	35.4	97.3	0.0	68.0	83.5	0.0		
CLR	82.8	41.7	97.3	0.0	71.4	84.3	0.0	80.3	35.4	95.7	0.0	60.7	82.9	0.0		
OneR	82.0	56.3	92.5	0.0	57.4	87.8	0.0	75.8	27.1	92.0	0.0	37.1	82.3	0.0		

remaining as the training dataset repeatedly for each instance of a given dataset, we can evaluate the performance of a learning algorithm to a new dataset without any ambiguity.

The results of the performances of the five learning algorithms to the entire training dataset and the results of Leave-One-Out are also shown in Table 3. All the Accuracies, Recalls of I and NI, and Precisions of I and NI are higher than those of the predicting default labels.

As compared to the accuracy of OneR, the other learning algorithms achieve equal or higher performances using combinations of multiple objective indices than by sorting with single objective index. With regard to the Recall values over class I, BPNN has achieved the highest performance. The other algorithms exhibit lower performance than that of OneR, because they tend to be learned classification patterns for the major class NI.

The accuracy of Leave-One-Out demonstrates the robustness of each learning algorithm. The Accuracy (%) of these learning algorithms ranges from 75.8% to 81.9%. However, these learning algorithms have not been able to classify the instances of class NU, because it is difficult to predict a minor class label in this dataset.

Learning Curves of the Learning Algorithms. Since the rule evaluation model construction method requires the mined rules to be evaluated by a human expert, we have investigated learning curves of each learning algorithm to estimate a minimum training subset to construct a valid rule evaluation model. The upper table in Fig. 2 shows the accuracies to the entire training dataset with each subset of training dataset. The percentage of achievements of each learning algorithm compared with their accuracy over the whole dataset are shown in the lower section of Fig. 2.

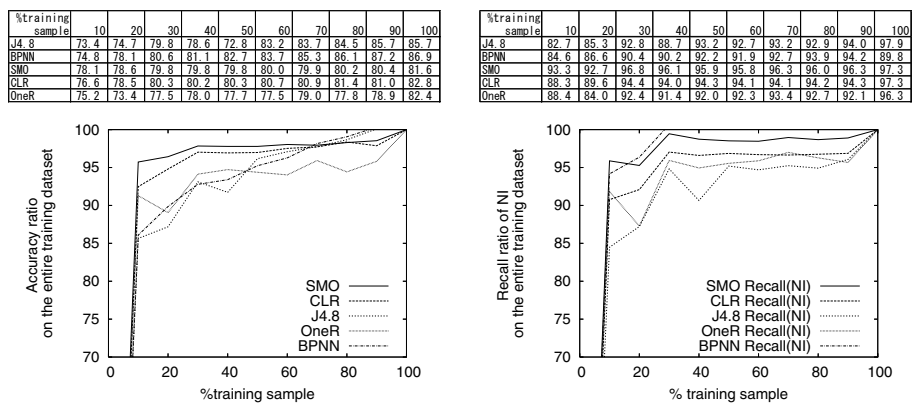


Fig. 2. Learning curves of Accuracies (%) on the learning algorithms over subsampled training dataset: The left table shows accuracies (%) of each training dataset to the entire dataset. The left graph shows their achievement ratios (%). The right table shows recalls (%) and the graph shows their achievement ratios(%).

As observed in these results, SVM and CLR, which learn hype-planes, obtained an achievement ratio grater than 95% using less than 10% of training subset. Although a decision tree learner and BPNN could learn better classifiers to the entire dataset than the hyper-plane learners, they need more training instances to learn accurate classifiers.

In order to eliminate known ordinary knowledge from a large rule set, the non-interesting rules need to be classified correctly. The right upper table in Fig. 2 shows percentage of Recalls on NI. The right lower chart in Fig. 2 also shows the percentage of achievements of Recall of NI, and compares it with the Recall of NI of the entire training dataset. From this result, we can eliminate the NI rules with rule evaluation models from SVM and BPNN although there only 10% of rule evaluations are conducted by a human expert. This fact is guaranteed with no less than 80% precisions for all learning algorithms.

Rule Evaluation Models on the Actual Datamining Result Dataset.

In this section, we present rule evaluation models for the entire dataset learned using OneR, J4.8 and CLR. This is because they are represented as explicit models such as a rule set, a decision tree, and linear model set.

Fig. 3 shows rule evaluation models for the actual data mining result: The rule set of OneR is shown in Fig. 3 (a), Fig. 3 (b) shows the decision tree learned with J4.8, and Fig. 3 (c) shows linear models used to classify each class.

As shown in Fig. 3 and Fig. 4, the indices used in the learned rule evaluation models are not only taken from a group of indices which increase with correctness of a rule, but also from different groups of indices. Indices such as YLI1, Laplace Correction, Accuracy, Precision, Recall, Coverage, PSI and, Gini Gain are indices which are formerly used on the models. The latter indices are GBI and Peculiarity, which sums up the difference in antecedents between one rule and the other rules in the same rule set. This corresponds to the comment provided

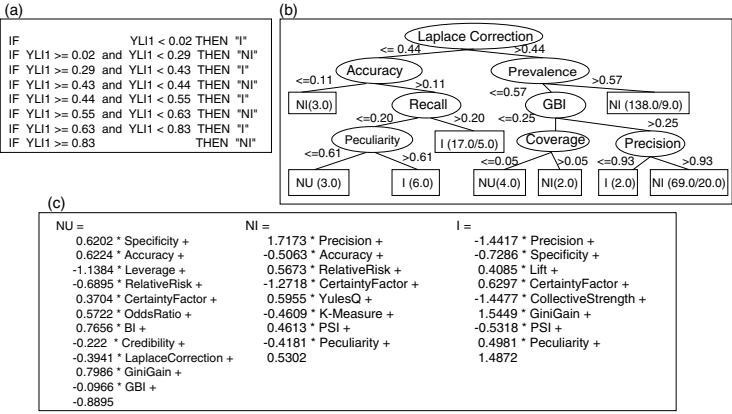


Fig. 3. Learned models for the meningitis data mining result dataset

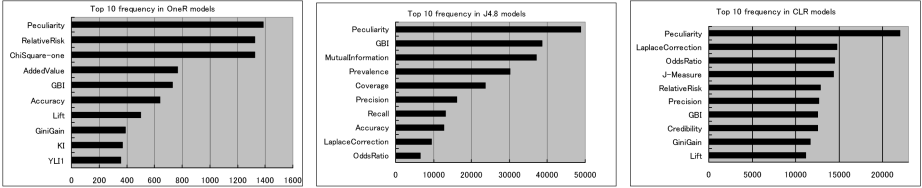


Fig. 4. Top 10 frequencies of the indices used by the models of each learning algorithm with 10000 bootstrap samples of the meningitis datamining result dataset and executions

by the human expert. He said that he evaluated these rules not only according to their correctness but also their interestingness based on his expertise.

3.2 Constructing Rule Evaluation Models on Artificial Evaluation Labels

We have also evaluated our rule evaluation model construction method using rule sets obtained from five datasets of the UCI machine learning repository to confirm the lower limit performances on probabilistic class distributions.

We selected the following five datasets: Mushroom, Heart, Internet Advertisement Identification (called InternetAd later), Waveform, and Letter. With these datasets, we obtained rule sets with bagged PART, which repeatedly executes PART [4] to the bootstrapped training subsample datasets.

For these rule sets, we calculated 39 objective indices as attributes of each rule. With regard to the classes of these datasets, we used three class distributions with multinomial distribution. Table 4 shows us the process flow diagram for obtaining these datasets and their description with three different class distributions. The class distribution for “Distribution I” is $P = (0.35, 0.3, 0.3)$ where p_i is the probability of class i . Thus, the number of class i instances in each dataset

Table 4. Flow diagram to obtain datasets and the datasets of the rule sets learned from the UCI benchmark datasets

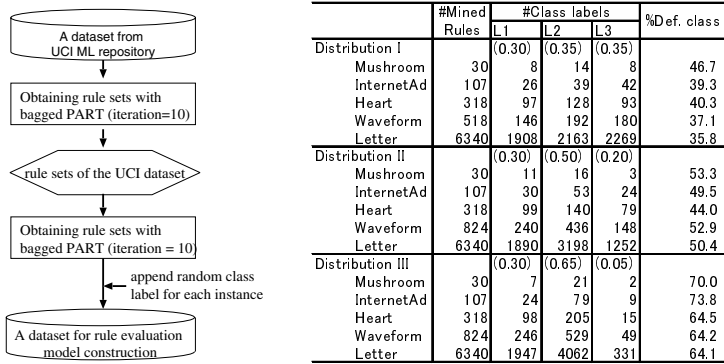


Table 5. Accuracies (%) on entire training datasets labeled with three different distributions (left table). Number of minimum training subsamples for outperforming the Accuracy (%) of default class (right table).

	J48	BPNN	SVM	CLR	OneR
Distribution I					
Mushroom	80.0	93.3	56.7	66.7	53.3
InternetAd	84.1	82.2	29.9	53.3	60.7
Heart	78.0	75.8	40.3	42.5	54.7
Waveform	46.5	46.4	37.6	39.8	54.9
Letter	36.8	36.4	30.1	36.6	52.1
Distribution II					
Mushroom	93.3	93.3	80.0	80.0	76.7
InternetAd	73.8	79.4	49.5	59.8	60.7
Heart	72.3	69.2	35.9	47.8	55.7
Waveform	61.2	57.8	52.9	53.0	59.7
Letter	51.0	51.0	50.4	50.4	57.0
Distribution III					
Mushroom	93.3	96.7	70.0	70.0	76.7
InternetAd	86.0	90.7	70.1	69.2	72.0
Heart	78.0	77.7	64.5	65.7	71.4
Waveform	74.4	69.3	64.2	64.2	69.3
Letter	64.1	64.3	64.1	64.1	68.3

	J48	BPNN	SVM	CLR	OneR
Distribution I					
Mushroom	8	8	12	18	14
InternetAd	14	14	-	30	14
Heart	42	31	66	114	98
Waveform	60	52	46	355	152
Letter	189	217	-	955	305
Distribution II					
Mushroom	6	4	4	6	12
InternetAd	24	24	52	42	70
Heart	52	40	-	104	92
Waveform	251	355	763	-	533
Letter	897	>1000	451	-	>1000
Distribution III					
Mushroom	22	14	22	28	22
InternetAd	80	66	-	-	-
Heart	114	94	142	318	182
Waveform	329	425	191	-	601
Letter	>1000	>1000	998	>1000	>1000

D_j become $p_i D_j$. Similarly, the probability vector of “Distribution II” is $P = (0.3, 0.5, 0.2)$ and that of “Distribution III” is $P = (0.3, 0.65, 0.05)$.

Accuracy Comparison on Classification Performances. To abovementioned datasets, we have used the five learning algorithms to estimate if their classification results reach or exceed the accuracies of that of just predicting each default class. The left table of Table 5 shows the accuracies of the five learning algorithms applied to each class distribution of the three datasets. As shown in Table 5, J4.8 and BPNN always perform better for just predicting a default class. However, their performances suffer from probabilistic class distributions for larger datasets such as Heart, Waveform and Letter.

Evaluation of Learning Curves. Similar to the evaluations of the learning curves on the meningitis rule set, we have estimated minimum training subsets for a valid model, which works better for just predicting a default class.

The right table in Table 5 shows the sizes of the minimum training subsets, which can help construct more accurate rule evaluation models than percentages of a default class formed by each learning algorithm. With smaller dataset such as Mushroom and InternetAd, these learning algorithms have been able to construct valid models with less than 20% of the given training datasets. However, to larger dataset, they need more training subsets to construct valid models, because their performances with whole training dataset fall to the percentages of default class of each dataset as shown in the left table in Table 5.

4 Conclusion

In this paper, we have described the evaluation of five learning algorithms for a rule evaluation support method with rule evaluation models to predict evaluations for an if-then rule based on objective indices by re-using evaluations by a human expert.

Based on the performance comparison of the five learning algorithms, rule evaluation models have achieved higher accuracies for just predicting each default class. Considering the difference between the actual evaluation labeling and the artificial evaluation labeling, it is shown that the evaluation of the medical expert considered particular relations between an antecedent and a class/another antecedent in each rule. By using these learning algorithms for estimating the robustness of a new rule with Leave-One-Out, we have achieved an accuracy greater than 75.8%. By evaluating learning curves, SVM and CLR were observed to have achieved an achievement ratio greater than 95% using less than 10% of the subset of the training dataset, which includes certain human evaluations. These results indicate the availability of this rule evaluation support method for a human expert.

In the future, we will introduce a selection method of learning algorithms to construct a proper rule evaluation model according to each situation. We also apply this rule evaluation support method to estimate other data mining result such as decision tree, rule set, and combination of them with objective indices, which evaluate all the mining results.

References

1. Ali, K., Manganaris, S., Srikant, R.: Partial Classification Using Association Rules. Proc. of Int. Conf. on Knowledge Discovery and Data Mining KDD-1997 (1997) 115–118
2. Brin, S., Motwani, R., Ullman, J., Tsur, S.: Dynamic itemset counting and implication rules for market basket data. Proc. of ACM SIGMOD Int. Conf. on Management of Data (1997) 255–264
3. Frank, E., Wang, Y., Inglis, S., Holmes, G., and Witten, I. H.: Using model trees for classification. Machine Learning, Vol.32, No.1 (1998) 63–76
4. Frank, E., Witten, I. H.: Generating accurate rule sets without global optimization. Proc. of the Fifteenth International Conference on Machine Learning, (1998) 144–151
5. Gago, P., Bento, C.: A Metric for Selection of the Most Promising Rules. Proc. of Euro. Conf. on the Principles of Data Mining and Knowledge Discovery PKDD-1998 (1998) 19–27

6. Goodman, L. A., Kruskal, W. H.: Measures of association for cross classifications. Springer Series in Statistics, 1, Springer-Verlag (1979)
7. Gray, B., Orlowska, M. E.: CCAIA: Clustering Categorical Attributes into Interesting Association Rules. Proc. of Pacific-Asia Conf. on Knowledge Discovery and Data Mining PAKDD-1998 (1998) 132–143
8. Hamilton, H. J., Shan, N., Ziarko, W.: Machine Learning of Credible Classifications. Proc. of Australian Conf. on Artificial Intelligence AI-1997 (1997) 330–339
9. Hatazawa, H., Negishi, N., Suyama, A., Tsumoto, S., and Yamaguchi, T.: Knowledge Discovery Support from a Meningoencephalitis Database Using an Automatic Composition Tool for Inductive Applications. Proc. of KDD Challenge 2000 in conjunction with PAKDD2000 (2000) 28–33
10. Hettich, S., Blake, C. L., and Merz, C. J.: UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mllearn/MLRepository.html>], Irvine, CA: University of California, Department of Information and Computer Science, (1998).
11. Hilderman, R. J. and Hamilton, H. J.: Knowledge Discovery and Measure of Interest. Kluwer Academic Publishers (2001)
12. Hinton, G. E.: “Learning distributed representations of concepts”, *Proceedings of 8th Annual Conference of the Cognitive Science Society*, Amherst, MA. REprinted in R.G.M.Morris (ed.) (1986)
13. Holte, R. C.: Very simple classification rules perform well on most commonly used datasets, Machine Learning, Vol. 11 (1993) 63–91
14. Klösgen, W.: Explora: A Multipattern and Multistrategy Discovery Assistant. in Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy R. (Eds.): Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, California (1996) 249–271
15. Ohsaki, M., Kitaguchi, S., Kume, S., Yokoi, H., and Yamaguchi, T.: Evaluation of Rule Interestingness Measures with a Clinical Dataset on Hepatitis. Proc. of ECML/PKDD 2004, LNAI3202 (2004) 362–373
16. Piatetsky-Shapiro, G.: Discovery, Analysis and Presentation of Strong Rules. in Piatetsky-Shapiro, G., Frawley, W. J. (eds.): Knowledge Discovery in Databases. AAAI/MIT Press (1991) 229–248
17. Platt, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: B. Schölkopf, C. Burges, and A. Smola (eds.): Advances in Kernel Methods - Support Vector Learning, MIT Press (1999) 185–208
18. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, (1993)
19. Rijsbergen, C.: Information Retrieval, Chapter 7, Butterworths, London, (1979) <http://www.dcs.gla.ac.uk/Keith/Chapter.7/Ch.7.html>
20. Smyth, P., Goodman, R. M.: Rule Induction using Information Theory. in Piatetsky-Shapiro, G., Frawley, W. J. (eds.): Knowledge Discovery in Databases. AAAI/MIT Press (1991) 159–176
21. Tan, P. N., Kumar V., Srivastava, J.: Selecting the Right Interestingness Measure for Association Patterns. Proc. of Int. Conf. on Knowledge Discovery and Data Mining KDD-2002 (2002) 32–41
22. Witten, I. H and Frank, E.: DataMining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, (2000)
23. Yao, Y. Y. Zhong, N.: An Analysis of Quantitative Measures Associated with Rules. Proc. of Pacific-Asia Conf. on Knowledge Discovery and Data Mining PAKDD-1999 (1999) 479–488
24. Zhong, N., Yao, Y. Y., Ohshima, M.: Peculiarity Oriented Multi-Database Mining. IEEE Trans. on Knowledge and Data Engineering, 15, 4, (2003) 952–960

Quality Assessment of k -NN Multi-label Classification for Music Data

Alicja Wiczorkowska and Piotr Synak

Polish-Japanese Institute of Information Technology
ul. Koszykowa 86, 02-008 Warsaw, Poland
{alicja, synak}@pjwstk.edu.pl

Abstract. This paper investigates problems related to quality assessment in the case of multi-label automatic classification of data, using k -Nearest Neighbor classifier. Various methods of assigning classes, as well as measures of assessing the quality of classification results are proposed and investigated both theoretically and in practical tests. In our experiments, audio data representing short music excerpts of various emotional contents were parameterized and then used for training and testing. Class labels represented emotions assigned to a given audio excerpt. The experiments show how various measures influence quality assessment of automatic classification of multi-label data.

1 Introduction

Automatic data classifiers, where a tested object is to be assigned to one of pre-defined classes, are broadly used worldwide and they are very useful in many applications. However, some data do not fit into this classification scheme. For instance, when listening to a piece of music from an audio database, one can feel various emotions, and when such data are classified with respect to these emotional states, multi-label classification is much more useful. In this case, each piece of music could be labeled with various emotions associated to this music. Therefore, we decided to investigate multi-label classification of data, and the problem of constructing a relevant classifier. Another problem to deal with is how the classification quality can be defined in the case of multi-label data. The K -Nearest Neighbor classifier was adopted to investigate all these issues.

2 Multi-label Classification

Multi-label classification has already been performed in numerous applications in text mining and scene classification domains, where documents or images can be labeled with several labels describing their contents [1], [2], [6]. Such a classification requires considering additional issues, including the selection of the training model, as well as set-up of testing and evaluation of results.

The following scenarios can be used for training of a classifier using examples with multiple labels:

- *MODEL-s* - the simplest model, assuming labeling of data by using single, the most likely label,
- *MODEL-i* - in this model, all the cases with more than one label are ignored, but in such a model data for training do not even exist in fully multi-labeled data set,
- *MODEL-n* - in this case, new classes are created for each combination of labels occurring in the training sample; however, in this model the number of classes easily becomes very large, especially, if the number of labels is only limited by the number of available labels, and for such a huge number of classes the data may easily become very sparse, so some classes may have very few training samples,
- *MODEL-x* - in this model a cross-training is performed, where samples with many labels are used as positive examples (and not as negative examples) for each class corresponding to the labels.

It could be interesting to experiment with all models, for example with *MODEL-n* and check how pairs or triplets of labels work [4]. However, in our experiments we decided to follow the *MODEL-x*, since we consider it to be the most efficient model for our data, where each tuple was labeled by several labels.

3 *K*-NN for Multi-label Classification

The research described in this paper was performed using *K*-Nearest Neighbor (*k*-NN) type of classification. In this chapter, we propose a *k*-NN classifier adapted to multi-label classification problem.

3.1 Classification Process

The standard *k*-NN classifier can be modified to suit multi-label data. First of all, in order to assign a set of labels to the tested object, the histogram presenting the number of appearances of each class label in the neighborhood is calculated. Then, two versions of the algorithm assigning class labels are proposed, and then also practically tested. In the first version, the assignment of classes for the tested object is performed on the basis of the measure $p_1 \in [0, 1]$. This measure is assigned to each label found in *k* nearest neighbors, and calculated in such a way that the histogram value for a label (i.e., number of neighbors where it appears) is normalized by the number of all labels, including repetitions, returned by the algorithm for a given testing sample (see Figure 1).

This p can be considered as a probability measure. The initial output of the classification algorithm is a set of labels with assigned probability p_1 , and only the labels exceeding some threshold level (chosen experimentally) are yielded as a final classification result. However, the global threshold is not the best choice, since the more labels assigned to the object, the lower probabilities are obtained. Therefore, a local threshold, chosen with respect to the obtained *k*-NN distribution, may yield better results.

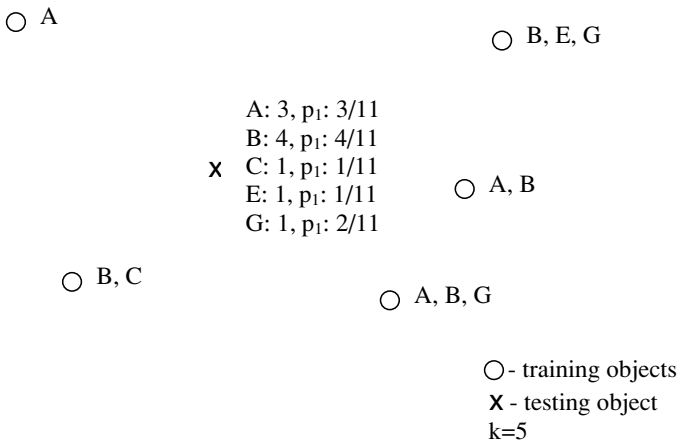


Fig. 1. Assignment of the measure p_1 for each class in the multi-class k -NN

Of interest is also to consider another measure p for assignment of class labels into the tested sample, or to consider a different threshold. There is particular reason for such consideration. Let us investigate 2 similar cases. If we have k -NN with $k = 6$, the tested object representing class A, and the following 6 results:

1. {A},
2. {A},
3. {A},
4. {A},
5. {A},
6. {A},

then we obtain $p_1=1$ for the class A and perfect classification. But on the other hand, if we have k -NN with $k = 6$, the tested object representing classes {A, B, C, D, E} and the following 6 results:

1. {A, B, C, D, E},
2. {A, B, C, D, E},
3. {A, B, C, D, E},
4. {A, B, C, D, E},
5. {A, B, C, D, E},
6. {A, B, C, D, E},

then the probability p_1 assigned to each class is equal to $1/5$. Therefore, k -NN again yields perfect match of class labels. However, if the threshold is high, then all these class labels can be rejected. This is why we decided to introduce a different measure, p_2 , calculated in such a way that the histogram value for each class was normalized by k , i.e. the number of neighbors (see Figure 2).

In this measure, all class labels are considered separately, and higher values of the measure are obtained. Therefore, the algorithm is not sensitive to the number of labels in k -neighborhood.

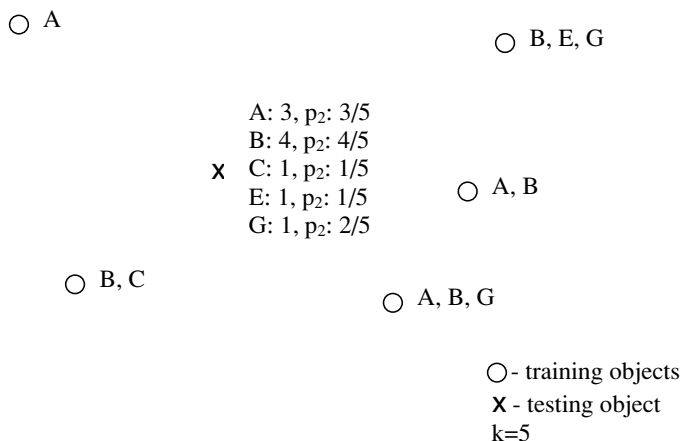


Fig. 2. Assignment of the measure p_2 for each class in the multi-class k -NN

3.2 Testing: Classification Accuracy Measurements

In order to assess the correctness of the obtained classification results, the measure of accuracy has to be defined.

Let X - set of labels for a tested object, found by the k -NN algorithm. Let Y - set of original labels for this object. Then, the classification accuracy acc for this object is defined as

$$acc = \frac{card(X \cap Y)}{card(X \cup Y)}.$$

For instance, if $X=\{A,B,C,E\}$ - labels found by k -NN, and $Y=\{A,B,D\}$ - original labels, then $acc = 2/5$. The final measure was averaged across the entire test set.

Of course, the definition of the measure determines the obtained accuracy level, so if another measure is defined, worse or better accuracy can be obtained, and better result may only mean that the measure is more strict, or not so strict with respect to the classification errors.

When defining the accuracy measure, it cannot depend on the method how the labels are chosen by the algorithm, so any of the probabilities (or neighbors) cannot be taken into account. The measure of the accuracy may only depend on the original labels and the labels assigned by the algorithm.

On one hand, one may want to consider separately precision and recall of the classifier, because the collective measure always depends on how false positives and omissions influence the final classification accuracy measure. Therefore, in the simplest way one can consider:

- precision=(number of labels correctly recognized by the algorithm)/(number of all classes assigned by the algorithm)
- recall=(number of labels correctly recognized by the algorithm)/(number of all classes actually labeling this object)

For instance, if $X=\{A, B, C\}$ - actual (original) labels of the object, $Y=\{A, B\}$ - labels found by the algorithm, then we have an omission. Therefore, although the precision for Y is equal to 1, the recall is equal to $2/3$. If the original labels are $X=\{A, B, C, D, E\}$, and the labels found by the algorithm are $Y=\{A, B, C, D, E, F\}$, then we have a false positive and the precision is equal to $5/6$, although the recall is equal to 1.

On the other hand, the total error for the entire test set is useful for assessing the quality of the classification. Therefore, the measure *acc* was used in this research, since it estimates accuracy of the entire classification process.

One can also use the most strict, binary measure, when 0 accuracy is assigned if the recognition result differs for a given object, and 1 when the labels found by the algorithm are identical with the original labeling. However, this would be unjust measure, since it seems to be too strict.

4 Binary Classification

Apart from classifying data in multi-class case, one can also consider binary classification, where each class is recognized against the rest of the classes, for each class separately. Any classifier can be applied for this purpose. In the training phase for a given class, all objects which are labeled with this class (and possibly with other classes) are considered as positive examples, whereas objects do not containing this class label are considered as negative examples. The weights can be also assigned to classification rules, to set up priorities of using them in the testing phase. For instance, rules against a given class can be assigned lower weight (or even discarded).

For such a binary classification, even for single-label classification the results can be much higher, since in this case the recognition accuracy mainly depends on the number of objects representing each class, and how big is the class in proportion to the entire data set. Generally, if the data set consists of numerous classes, then even the classifier that always votes against the tested class has the accuracy level equal to $1-(\text{number of objects in the testes class})/(\text{number of objects in the entire data set})$. Therefore, high recognition accuracy is easy to obtain in this case, but it does not imply good discernment power of the classifier. Instead, one should rather consider general recognition. Since one is usually interested in checking how the results reflect the classification abilities of the algorithm and descriptors, one should focus on the issue how various improvements influence final classification quality, i.e. if it has been improved or deteriorated. However, the binary recognition evokes a few interesting issues on how to assign class labels and estimate classification results.

For example, in case of multi-label classification one must consider the following situation. If the object $X=\{A, B, D\}$ is used in training for the class A (against the other classes), how to assess the accuracy, if the algorithm recognizes it as B , or as D ? One of the possibilities is to introduce additional recognition class, neutral, and assign it in this case, i.e. consider the result neither correct

nor incorrect (but then we risk "neutral" assessment even for a perfect classifier, for instance showing $\{A, B, D\}$ in this case); if other classes are assigned by the algorithm, the accuracy should be equal to zero. Of course, the same object labeled by the algorithm as $\{A, B, D\}$ should be assessed as correctly classified. So, if one performs the experiment for the class A, then yielded labeling that includes this class should be considered correct. However, all class labels not included in the original labeling of the test object (for instance G in this case) should lower the final accuracy measure. Also, if the classifier yields $X=\{A, B\}$ for this case, the accuracy measure should be lower. The final measure should consider punishment for wrong labels assigned to the tested object. Also, the accuracy estimation of the algorithm should consider the fact, that the object labeled $\{A, B\}$ is different than the object labeled $\{A\}$. For instance, if the object $\{A, B, C, D, E, F\}$ is tested, it is difficult to decide which class it should actually represent; if the test for the A class against the rest is performed, then this particular object may be considered as representing other classes to higher extent than it represents class A. On the other hand, if the classifier assigns $\{A, G, H, I, J, K, L, M\}$ to this object, then it is still considered correct, even if it is almost totally wrong. One can even consider each combination of labels as a separate super-label, but this approach would lead to numerous classes with very few representants, not of much use for experiments.

Generally, the measure should take into account the length of original labeling of the object and the length of the labeling found by the classification algorithm (and possibly the number of available classes). In standard setting, the weights for both omissions and false positives should be the same. One can also observe which classes are most difficult for the classifier, i.e. for which classes the classifier is most frequently wrong, and with which classes the objects are mistaken.

5 Experiments

The theoretical issues investigations in the previous sections has been practically tested on the data representing audio sequences, labeled with emotions they evoke when human subjects listen to them.

5.1 Data

The data used in experiments represented 875 audio excerpts from musical recordings. Each audio fragment was 30 second long, recorded stereo in .mp3 format and then converted to .au format, with 44100 Hz sampling frequency and 16-bit resolution. Spectral analysis was performed on 32768 samples long analyzing frame, taken from the left channel, with Hanning window applied. The spectral components up to 12 kHz and no more than 100 partials were taken into account when calculating parameters describing various aspects of long-time spectrum. The following 29 parameters constitute the feature vector [8], [9]:

- *Frequency*: dominating fundamental frequency,
- *Level*: maximal level of sound,

- *Tristimulus*_{1,2,3}: Tristimulus parameters for *Frequency*, describing strength of the fundamental, the middle partials (no. 2, 3, and 4) and high partials in the spectrum [7],
- *EvenHarm* and *OddHarm*: Contents of even and odd harmonics in the spectrum,
- *Brightness*: brightness of sound - gravity center of the spectrum,
- *Irregularity*: irregularity of spectrum (see [3]),
- *Frequency*₁, *RatioF*₁, ..., 9: parameters describing 10 most prominent peaks in the spectrum. The lowest frequency within this set is chosen as *Frequency*₁, and proportions of other frequencies to the lowest one are denoted as *RatioF*₁, ..., 9,
- *Amplitude*₁, *RatioA*₁, ..., 9: the amplitude of *Frequency*₁ in decibel scale, and differences in decibels between peaks corresponding to *RatioA*₁, ..., 9 and *Amplitude*₁.

The data represent various emotions, perceived by a human when listening to a given excerpt. The data were labeled by musicians, using 13 classes [5]:

1. frustrated,
2. bluesy, melancholy,
3. longing, pathetic,
4. cheerful, gay, happy,
5. dark, depressing,
6. delicate, graceful,
7. dramatic, emphatic,
8. dreamy, leisurely,
9. agitated, exciting, enthusiastic,
10. fanciful, light,
11. mysterious, spooky,
12. passionate,
13. sacred, spiritual.

Each tuple was initially labeled with a single emotion, perceived while listening to the audio excerpt represented by the given tuple, and then labeled again, with no limitation to the number of assigned labels. Number of objects assigned to each class is presented in Table 1.

The unbalance in the number of elements in the different classes could be removed by, for instance, stratification. In our research, the grouping of data was redefined - the classes were combined and 6 super-classes were created [5]:

1. frustrated, agitated, exciting, enthusiastic, dramatic, emphatic,
2. bluesy, melancholy, dark, depressing,
3. longing, pathetic, passionate,
4. cheerful, gay, happy, fanciful, light,
5. delicate, graceful, dreamy, leisurely,
6. mysterious, spooky, sacred, spiritual.

Number of classes assigned to each of 6 super-classes is shown in Table 2.

Table 1. Number of objects representing emotion classes in 875-element multi-labeled database of 30-second audio samples, for grouping into 13 classes

Class	No. of objects	Class	No. of objects
Agitated, exciting, enthusiastic	304	Fanciful, light	317
Bluesy, melancholy	214	Frustrated	62
Cheerful, gay, happy	62	Longing, pathetic	147
Dark, depressing	41	Mysterious, spooky	100
Delicate, graceful	226	Passionate	106
Dramatic, emphatic	128	Sacred, spiritual	23
Dreamy, leisurely	151		

Table 2. Number of objects representing emotion classes in 875-element multi-labeled database of 30-second audio samples, for grouping into 6 super-classes

Class	No. of objects
Agitated, exciting, enthusiastic Dramatic, emphatic Frustrated	494
Bluesy, melancholy Dark, depressing	255
Cheerful, gay, happy Fanciful, light	379
Delicate, graceful Dreamy, leisurely	377
Longing, pathetic Passionate	253
Mysterious, spooky Sacred, spiritual	123

5.2 Results

The experiments started with single-label data. For 13 classes the obtained accuracy was 20.12%, for $k = 13$. For 6 super-classes, 37.47% correctness was obtained, for $k = 13$. Therefore, the results were low. However, subjective tests conducted with human experts (experienced musicians) also gave low results, since compatibility of classification among experts was at the level of 24-33%.

Next, experiments for multi-labeled data were performed. Firstly, the probability measure p_1 was used, with the threshold value equal to 0.15 yielding the best results. For 13 classes, 27.1% correctness for $k = 13$ was obtained, and for 6 super-classes, 38.62% correctness for $k = 15$, using the accuracy measure acc .

For the probability measure p_2 , $k=15$ yielded the best results. The accuracy 28.05% was obtained for 13 classes and the threshold value equal to 0.28. For 6 super-classes, accuracy 38.98% was obtained for the threshold value 0.32.

In case of binary classification, even for single-label classification results were much higher, as illustrated in Figure 3. Like previously, the k -NN algorithm was applied in the experiments. In case of such a classification the recognition accuracy mainly depends on the number of objects representing each class (and how big is the class in proportion to the entire data set). For the smallest class, the results have even reached 95%.

Class	No. of objects	Correctness
1. happy, fanciful	74	95.97%
2. graceful, dreamy	91	89.77%
3. pathetic, passionate	195	71.72%
4. dramatic, agitated, frustrated	327	64.02%
5. sacred, spooky	88	89.88%
6. dark, bluesy,	97	88.80%

Fig. 3. Results of binary classification of emotions for 6 super-classes

The results presented in Figure 3, although tempting, are meaningless, since they can be easily obtained without any classifier, only by pointing always against the tested class (as mentioned in Section 4).

In case of the specific data used in the described experiments, one must consider yet another problem. Namely, the emotions in a 30-second music piece may change, even a few times within this time, so probably more detailed labeling (considering changes of emotions in time) should be performed on these data before further experiments.

For all the reasons mentioned above, we did not perform further experiments with binary classification, neither for single nor multi-labeling.

6 Summary

In this paper, we investigated theoretical problems related to multi-label classification of data, and tested them practically on database of multi-labeled tuples, where each datum could have been labeled by any number of 6 or 13 classes, representing emotions perceived by human subjects when listening to the parameterized music pieces. K -NN classifier was used in our research. The result of this work is a methodology for multi-label classification of data, illustrated with practical experiments using data representing audio files. Although the accuracy of this classification was not high, it still outperformed single-label accuracy, and was comparable with human performance. The proposed methodology can also be applied to other types of data.

Acknowledgements

This research was supported by the grant 3 T11C 002 26 from Ministry of Scientific Research and Information Technology of the Republic of Poland, by the National Science Foundation under grant IIS-0414815 and by the Research Center at the Polish-Japanese Institute of Information Technology, Warsaw, Poland.

The authors would like to express thanks to Doctor Rory A. Lewis from the University of North Carolina at Charlotte for elaborating the initial audio database for research purposes, and to Doctor Zbigniew W. Raś from the UNCC for numerous discussions regarding binary multi-label classification.

References

1. Boutell, M., Shen, X., Luo, J., Brown, C.: Multi-label Semantic Scene Classification. Technical Report, Dept. of Computer Science, U. Rochester (2003)
2. Clare, A., King, R.D.: Knowledge Discovery in Multi-label Phenotype Data. *Lecture Notes in Computer Science* **2168** (2001) 42–53
3. Fujinaga, I., McMillan, K.: Realtime recognition of orchestral instruments. *Proceedings of the International Computer Music Conference*, (2000) 141–143.
4. Ghamrawi, N., McCallum, A.: Collective Multi-Label Classification. *Fourteenth Conference on Information and Knowledge Management*, (2005) 195–200.
5. Li, T., Ogihara, M.: Detecting emotion in music. *4th International Conference on Music Information Retrieval ISMIR*, Washington, D.C., and Baltimore, MD. (2003)
6. McCallum, A.: Multi-label Text Classification with a Mixture Model Trained by EM. *AAAI'99 Workshop on Text Learning* (1999)
7. Pollard, H. F., Jansson, E. V.: A Tristimulus Method for the Specification of Musical Timbre. *Acustica* **51** (1982) 162–171
8. Synak, P., Wieczorkowska, A.: Some Issues on Detecting Emotions in Music. In: D. Slezak, J. Yao, J. F. Peters, W. Ziarko, X. Hu (Eds.), *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*. 10th Int. Conf. RSFDGrC 2005, *Proceedings*, Part II. LNAI 3642, Springer, (2005) 314–322
9. Wieczorkowska, A., Synak, P., Lewis, R., Ras, Z. W.: Creating Reliable Database for Experiments on Extracting Emotions from Music. In: M. A. Kłopotek, S. Wierzhon, K. Trojanowski (Eds.), *Intelligent Information Processing and Web Mining. Proceedings IIPWM'05. Advances in Soft Computing*, Springer (2005) 395–402

Effective Mining of Fuzzy Multi-Cross-Level Weighted Association Rules

Mehmet Kaya¹ and Reda Alhajj^{2,3}

¹ Dept. of Computer Eng, Firat University, Elazig, Turkey

² Dept. of Computer Science, University of Calgary, Calgary, Alberta, Canada
alhajj@cpsc.ucalgary.ca

³ Dept. of Computer Science, Global University, Beirut, Lebanon

Abstract. This paper addresses fuzzy weighted multi-cross-level association rule mining. We define a fuzzy data cube, which facilitates for handling quantitative values of dimensional attributes, and hence allows for mining fuzzy association rules at different levels. A method is introduced for single dimension fuzzy weighted association rules mining. To the best of our knowledge, none of the studies described in the literature considers weighting the internal nodes in such taxonomy. Only items appearing in transactions are weighted to find more specific and important knowledge. But, sometimes weighting internal nodes on a tree may be more meaningful and enough. We compared the proposed approach to an existing approach that does not utilize fuzziness. The reported experimental results demonstrate the effectiveness and applicability of the proposed fuzzy weighted multi-cross-level mining approach.

1 Introduction

Data mining is a process for discovering previously unknown and potentially useful abstractions from the content of large databases. Mining association rules is one of the most intensively studied models for data mining. The basic task in mining for association rules is to determine the correlation between items belonging to a transactional database. In general, every association rule must satisfy two user specified constraints, one is support and the other is confidence. The support of a rule $X \rightarrow Y$ is defined as the fraction of transactions that contain $X \cup Y$, where X and Y are sets of items from the given database. The confidence is defined as the ratio $\frac{\text{support}(X \cup Y)}{\text{support}(X)}$. So, the target is to find all association rules that satisfy user specified minimum support and confidence values.

None of the data cube based approaches encountered in the literature, e.g. [1, 4–7, 13, 15], include quantitative attributes with fuzzified and weighted domains, which are more natural and understandable by humans. They mostly use data cubes with binary attributes or data cubes that discretize the domains of their attributes. This motivated the development of the novel method described in this paper. We contribute to the ongoing research on multidimensional online data mining by proposing general architecture that constructs and uses fuzzy data

cube for knowledge discovery. The idea behind introducing the fuzzy data cube architecture for online mining is to allow users to query a given database for fuzzy association rules based on different values of support and confidence. Described in [11] is one method that utilizes a fuzzy data cube for multi-dimensional fuzzy association rules mining. On the other hand, the method described in this paper is dedicated to one-dimensional fuzzy association rules mining. To the best of our knowledge, this is the first attempt to utilize fuzziness in OLAP mining. The method described in this paper has been tested using a subset from the adult data of the United States census in year 2000. Also, the proposed method has been compared with the discrete approach of Srikant and Agrawal [14]. The test results reported in this paper demonstrate the effectiveness and applicability of the proposed mining approach.

The rest of this paper is organized as follows. Section 2 covers the background and the basic terminology used in the rest of the paper. Section 3 presents the proposed fuzzy weighted multi-cross-level mining method. Experimental and comparison results are reported and discussed in Section 4. Section 5 is summary and conclusions.

2 Background and Basic Terminology

2.1 Fuzzy Data Cube Architecture

At least two fuzzy sets may be defined for each quantitative attribute x , with a membership function per fuzzy set, such that each value of x qualifies to be with a certain degree of membership in one or more of the fuzzy sets specified for x . Using fuzzy sets specified for different attributes, it is possible to construct a fuzzy data cube.

Definition 1 (Degree of Membership). *Given an attribute x and let $F_x = \{f_x^1, f_x^2, \dots, f_x^l\}$ be a set of l fuzzy sets associated with x . Each fuzzy set f_x^j in F_x has a corresponding membership function, denoted $\mu_{f_x^j}$, which represents a mapping from the domain of x into the interval $[0,1]$. Formally, $\mu_{f_x^j} : D_x \rightarrow [0,1]$.*

For a given value v of x , if $\mu_{f_x^j}(v) = 1$ then v totally and certainly belongs to fuzzy set f_x^j . On the other hand, $\mu_{f_x^j}$ means that v is not a member of fuzzy set f_x^j . All other values between 0 and 1, exclusive, specify a “partial membership” degree for v .

Definition 2 (Fuzzy Data Cube). *Consider a database with a number of quantitative attributes. It is possible to construct a corresponding fuzzy data cube with n dimensions $d_1, d_2, \dots, d_n$, such that each given quantitative attribute appears in only one dimension of the cube and each dimension d_j of the cube contains $\sum_{i=1}^k (l_i + 1)$ slots, where k is the number of attributes in dimension d_j , l_i is the number of membership functions for attribute x_i in dimension d_j , and “+1” in the formula represents a special slot named “Total”, which stores the aggregation values of the previous slots. These aggregation values show one of the essential features of the fuzzy data cube structure.*

2.2 Fuzzy Association Rules

An association rule is the correlation between data items; it is classified as either binary or quantitative. Quantitative association rules are defined over quantitative and categorical attributes. The work described in [14] maps the values of categorical attributes into set of consecutive integers and the values of quantitative attributes are first partitioned into intervals using equi-depth partitioning, if necessary, and then mapped into consecutive integers to preserve the ordering of the values/intervals. This way, both categorical and quantitative attributes can be handled in a uniform fashion as a set of $\langle \text{attribute}, \text{integer value} \rangle$ pairs. Based on the mapping defined in [14], a quantitative association rule is mapped into a set of boolean association rules. After the mapping, the algorithm for mining boolean association rules is applied to the transformed data set.

However, the intervals used in mining quantitative association rules may not be concise and meaningful enough for human experts to obtain nontrivial knowledge. Fuzzy sets provide smooth transition between members and non-members of a set; and fuzzy association rules are considered as convenient tool for handling quantitative attributes in human understandable manner because of the linguistic terms associated to fuzzy sets.

To elaborate on fuzzy association rules, consider a database of transactions T , its set of attributes I , and the fuzzy sets associated with quantitative attributes in I . Notice that each transaction t_i contains values of some attributes from I and each quantitative attribute in I is associated with at least two fuzzy sets. The target is to find out some interesting and potentially useful regularities, i.e., fuzzy association rules with enough support and high confidence. We use the following form for fuzzy association rules [12].

Definition 3. [*Fuzzy Association rules*] A fuzzy association rule is defined as:

If $X = \{x_1, x_2, \dots, x_p\}$ is $A = \{f_1, f_2, \dots, f_p\}$ then $Y = \{y_1, y_2, \dots, y_q\}$ is $B = \{g_1, g_2, \dots, g_q\}$,

where X, Y are itemsets, i.e., disjoint subsets of I , and A and B contain fuzzy sets associated with corresponding attributes in X and Y , respectively, i.e., f_i is a fuzzy set related to attribute x_i and g_j is a fuzzy set related to attribute y_j .

2.3 Multi-Cross-Level Association Rules Mining

Benefiting from the fuzzy data cube structure, we take the model one step further to deal with multi-cross-level mining in a fuzzy data cube. In such case, large itemsets can include nodes from more than one level. As only terminal nodes appear in transactions and since each level has different minimum support threshold, the computation of support thresholds of such itemsets is not trivial; it is more complicated than those of non-cross-level itemsets. Support thresholds of cross-level itemsets are computed as follows.

Proposition 1. Let $A(A_1, A_2, \dots, A_q)$ be an itemset. By definition, items in an itemset cannot be ancestors or descendants of each other. The support threshold of itemset A is: $\min_{i=1}^q (FMinSup(A_i))$, where $FMinSup(A_i)$ is the minimum fuzzy support value of level of A_i .

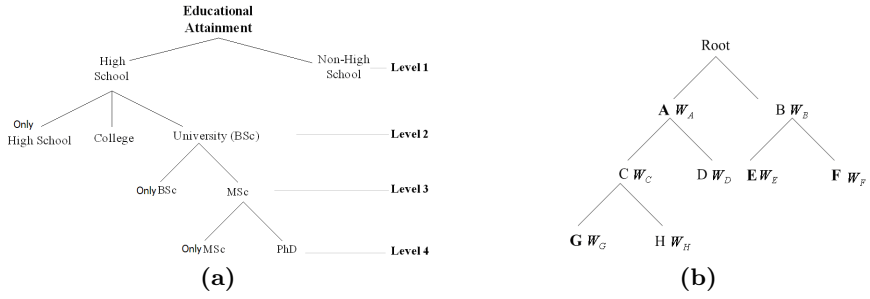


Fig. 1. a) Possible taxonomy breakdown of Educational Attainment dimension; b) Example weighted taxonomic structure

To illustrate items at different levels, consider possible classification of education degrees and the schooling system. Figure 1-a shows the hierarchy for the dimension *Educational Attainment*, where terminal nodes represent actual items appearing in transactions and internal nodes represent classes or concepts. For instance, internal node *High School* is at the first level (Level 1) in Figure 1-a; internal node *University* is at the second-level (Level 2); internal node *MSc* is at the third-level (Level 3); and terminal node *PhD* is at the fourth-level (Level 4).

Weighting Items at Multiple Concept Levels. In the approach proposed in this paper, we also realized the importance of assigning weights to items at multiple cross levels. Actually, the algorithms developed earlier to mine weighted association rules handle the single-concept level, e.g., [2, 16], i.e., only items appearing in transactions are weighted to find more specific and important knowledge. However, we argue that it is necessary to specify weight values for classes or concepts at higher levels instead of actual items present in transactions. This is true because sometimes weighting internal nodes in a tree may be more meaningful and enough, while another time both an ancestor and its descendant may need to be weighted. Also, weighting of nodes at different levels adds an important measure of interestingness. Doing this permits users to control the mining results because users' intuition is included in the association rules mining process. Finally, we handle weighting at multiple levels as follows.

Proposition 2. Consider $m + 1$ items, say $x, y_1, y_2, \dots, y_m$; and assume that x is ancestor of $y_1, y_2, \dots, y_m$, i.e., the latter items are descendants of x . While weighting nodes at multiple concept levels, the following cases that may be observed:

1. If only ancestor x is weighted, then all descendants of x get the same weight as x .
2. If the weights of all descendants of x are pre-specified, then the weight of x is computed as follows:

$W_x = \frac{\sum_{j=1}^m (TotalCount_{y_j} \cdot W_{y_j})}{TotalCount_x}$ where $TotalCount_{y_i} = \sum_{i=1}^n (v_{iy_j})$, here n is the number of transactions and v_{iy_j} is the value from domain D_i of descendant y_j .

3. If the weights of ancestor x and $m - 1$ of its descendants are given (assume the weight of descendant y_p is not given) then the weight of descendant y_p is computed with respect to the given weights:

$$W_{y_p} = \frac{TotalCount_x \cdot W_x - \sum_{j=1, j \neq p}^m (TotalCount_{y_j} \cdot W_{y_j})}{TotalCount_{y_p}}$$

Shown in Figure 1-b is an example tree to illustrate how the proposed method weights relevant nodes using predefined taxonomy. The weighted taxonomy tree shown in Figure 1-b has 3-levels. As the weights of the nodes denoted by bold letter were pre-specified, the weights of the others are computed by employing our approach discussed above. For instance, first the weight of node A is given as FW_A , so the weights of its descendants C and D are the same as that of A (Case 1). Second, if the weights of nodes E and F are pre-specified as FW_E and FW_F , respectively, then the weight of node B is determined as FW_B (Case 2). Third, after the weight of node C was found as FW_C and the weight of node G was given as FW_G , the weight of node H is computed as FW_H (Case 3).

Table 1. An example 2-dimensional fuzzy data cube

			EDUCATIONAL ATTAINMENT Dimension																		Total
			High School Education															Non-High School			
			Only High School			College			University Education												
									MSc						Only BSc						
									PhD			Only MSc									
			low	medium	high	low	medium	high	low	medium	high	low	medium	high	low	medium	high	low	medium	high	
Dimension CITIZENSHIP	American	White	15.2	5.1	3.7	11.4	5.7	0	0.75	0.25	0	12.6	7.1	1.1	24.7	8.3	36.7	21.2	12.7	4.1	170.6
		Black	8.5	4.7	10.2	7.4	1.8	0	2.3	5.5	0	5.7	4.3	0	14.7	5.5	25.1	12.7	15.1	7.8	131.3
		Other	6.4	0	8.1	6.7	2.3	0.8	4.3	2.7	1.2	3.9	1.8	0	25.2	2.8	40.3	13.1	24	6.7	150.3
	Non-American	White	5.3	7.8	3.7	5.1	5.7	4.3	7.8	4.8	0	8.4	5.4	2.7	9.4	4.9	31.7	3.8	9.8	5.5	126.1
		Black	4.2	11.4	2.8	2.0	6.8	1.4	5.4	5.1	0.7	3.0	4.2	1.4	14.0	6.7	28.2	7.9	10.7	2.7	118.6
		Other	6.8	6.7	5.7	4.3	8.5	3.3	3.3	1.4	2.4	2.7	0.8	0	14.2	3.8	34.0	5.5	6.4	4.0	113.8
	Total			46.4	35.7	34.2	36.9	30.8	9.8	23.85	19.75	4.3	36.3	23.6	5.2	102.2	32	196	64.2	78.7	30.8

Table 2. The transaction table generated from Table 1

Dimension CITIZENSHIP (Transaction Dimension)	Dimension to mine the associations among its attributes (EDUCATIONAL ATTAINMENT)
White	{(H, l), (H, m), (H, h), (C, l), (C, m) (P, l), (P, m), (M, l), (M, m), (M, h), (B, l), (B, m), (B, h), (N, l), (N, m), (N, h)}
Black	{(H, l), (H, m), (H, h), (C, l), (C, m) (P, l), (P, m), (M, l), (M, m), (B, l), (B, m), (B, h), (N, l), (N, m), (N, h)}
Other	{(H, l), (H, h), (C, l), (C, m), (C, h), (P, l), (P, m), (P, h), (M, l), (M, m), (B, l), (B, m), (B, h), (N, l), (N, m), (N, h)}
White	{(H, l), (H, m), (H, h), (C, l), (C, m), (C, h), (P, l), (P, m), (M, l), (M, m), (M, h), (B, l), (B, m), (B, h), (N, l), (N, m), (N, h)}
Black	{(H, l), (H, m), (H, h), (C, l), (C, m), (C, h), (P, l), (P, m), (P, h), (M, l), (M, m), (M, h), (B, l), (B, m), (B, h), (N, l), (N, m), (N, h)}
Other	{(H, l), (H, m), (H, h), (C, l), (C, m), (C, h), (P, l), (P, m), (P, h), (M, l), (M, m), (B, l), (B, m), (B, h), (N, l), (N, m), (N, h)}

3 The Proposed Multi-Cross-Level Fuzzy Weighted Association Rules Mining Approach

Single dimension association rules mining concentrates on the correlation within one dimension of a data cube by grouping the other dimensions. Shown in Table 1 is an example data cube, where two dimensions are involved in the single dimension association rules mining process. The developed system has an OLAP server, which is responsible for creating the target data cube. Created by the OLAP server, a two dimensional data cube serves as the input fuzzy data cube for the online mining process. One of the dimensions is referred to as transaction dimension and the other dimension is the set of attributes associated with the transaction dimension.

By grouping attributes in the transaction dimension, a transaction table can be transformed into a set-based table in which items sharing the same transactional attribute are merged into one tuple. Note that there is no point in partitioning attributes in the transaction dimension into fuzzy sets because the mining process is performed in the non-transaction dimension as illustrated in Table 2, where each attribute-fuzzy set pair is abbreviated for the sake of space. For instance, (H,l) and (C,m) stand for (only High school, low) and (College, medium), respectively.

Proposition 3. *Consider a tuple t from the relational table that represents a data cube; if the value of attribute y_j in t has membership degree $\mu^n(y_j)$ in the fuzzy set f_j^n , then the sharing rate, denoted SR , of the fuzzy set (y_j, f_j^n) to attribute x_i in the data cube is $\mu^n(y_j)$.*

The sharing rate is actually the production of membership values of the corresponding fuzzy sets. Proposition 3 is true because one of the dimensions is referred to as the transaction dimension in a cube with two dimensions. The membership grades for a given value of an attribute show the sharing rates of the relevant cells.

After constructing the fuzzy data cube, the process of association rules mining starts by identifying large itemsets. An itemset is large if it is a combination of items that have a support over a predefined minimum support. The mining process, and hence identifying large itemsets is performed based on the sum of sharing rates SR , which is introduced in terms of the degree of membership in the fuzzy sets. If more than one membership function intersect with the real values of an attribute, then all the cells related to these functions are updated. This way, each cell in Table 1 stores SR value.

To generate fuzzy association rules, all sets of items that have support above a user specified threshold should be determined first. Itemsets with support value greater than or equal to the minimum support value are called frequent or large itemsets. The following formula is used in calculating the fuzzy support value, $FSupport(Y, F)$, for an itemset Y in the item dimension and its corresponding set of fuzzy sets F :

$$FSupport(Y, F) = \frac{\sum_{t_i \in T} \prod_{(y_j \in Y)} (SR(y_j, f_j^n))}{|T|}$$

where $|T|$ is the number of transactions, t_i represents the i -th transaction in T , such as the attribute *White American* in the transaction dimension *Citizenship*, and y_j is an item in Y ; $\langle PhD, Medium \rangle$ is an example of y_j .

Another advantage of a fuzzy data cube is to be able to find interesting fuzzy rules at multiple levels.

4 Experimental Results

We performed some empirical tests on a real-life data in order to evaluate the effectiveness and applicability of the proposed approach and to analyze its scalability. We have used the real dataset to construct and study dense fuzzy data cube, and we have analyzed the scalability by constructing sparse fuzzy data cube. All the experiments were conducted on a Pentium IV 2GHz CPU with 512 MB of memory and running Windows XP. As the dataset is concerned, we constructed the fuzzy data cube using 8 attributes and 100K transactional records from the adult data of the United States census in year 2000.

For the experiments, we constructed a fuzzy data cube which complies with the requirements of the method described in Section 3; for each dimension of the cube, a set of attributes was picked from the experimental census database.

In all the experiments conducted for this part of the study, two different cases were considered as the per attribute number of fuzzy sets (FS) is concerned, namely 3 and 4 fuzzy sets, denoted FS3 and FS4, respectively. Also, in order to show the effectiveness of the fuzzy data cube over the traditional data cube in association rules mining, we compared our fuzzy approach with the discrete method proposed by Srikant and Agrawal [14]. For this purpose, we distributed the values of each quantitative attribute into 3 and 4 intervals, denoted Discrete3 and Discrete4, respectively. Finally, four tests are conducted to evaluate our approach with respect to the following dimensions: 1) number of large itemsets generated for different values of minimum support; 2) number of association rules generated for different values of minimum confidence; 3) number of association rules as a function of the average weight; and 4) execution time. Here, note that in all the experiments, unless otherwise specified, the minimum support value has been set to 15% for level 1, 5% for level 2 and 2% for level 3.

The experiments test the performance of the proposed method on 2-dimensional fuzzy data cube with 8 attributes used to construct the non-transaction dimension; the results are shown in Figures 2-4. Figure 2-a compares the number of large itemsets obtained for different values of minimum support only at level 3. As the value of minimum support increases, the number of large itemsets decreases. On the other hand, the curves that correspond to fuzzy sets extract higher number of large itemsets than those of the discrete method. Another phenomenon that can be deduced from the curves plotted in Figure 2-a is that the numbers of large itemsets decrease along with the increase in the number of fuzzy sets and discrete intervals. This is an expected situation because a large number of fuzzy sets or discrete intervals will lead to quantities of an

item in different transactions easily scattered. So, the total support of an item or itemset is reduced; and this is reflected into decrease in the number of large itemsets. Another important point is that the gap between the curves with the discrete method is larger than the gap between the curves of the fuzzy sets. This means that the number of large itemsets extracted from a fuzzy data cube is less sensitive to the increase in the number of fuzzy sets as compared to the number of large itemsets extracted from a traditional data cube with discrete intervals.

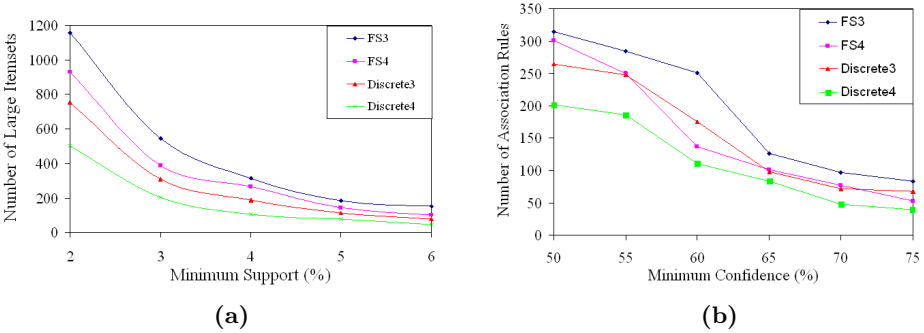


Fig. 2. a) Number of large itemsets vs. minimum support; b) Number of association rules vs minimum confidence

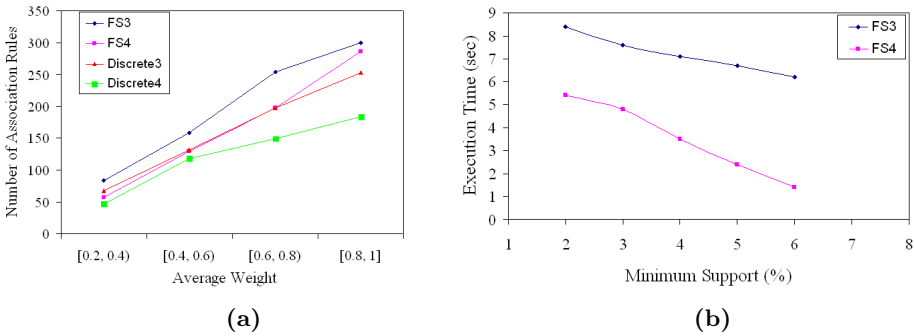


Fig. 3. a) Number of association rules vs random weight intervals; b) Execution time as minimum support changes

The curves plotted in Figure 2-b show the change in the number of association rules for different values of minimum confidence and for different numbers of fuzzy sets and discrete intervals. In a way similar to that of Figure 2-a, the number of association rules decreases as the number of fuzzy sets and discrete intervals increases. Also, the number of rules represented by each of the four curves decreases as the value of the minimum confidence increases. This is quite consistent with our intuition. Finally, it can be easily seen that the curve labeled

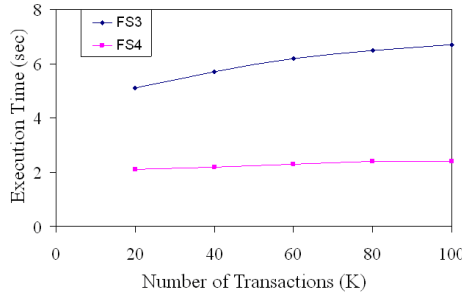


Fig. 4. Demonstrating scalability

Discrete4 is smoother than all the other curves, i.e., the minimum confidence values have large effect on the number of rules mined from the curves with fuzzy sets.

The next experiment deals with the increase in the number of rules as a function of the average weight; the results are reported in Figure 3-a, where the minimum confidence is set as 50%. We used four intervals, from which random weights were generated. The increase in the number of rules is almost linear with respect to the average weight.

In the fourth experiment, we examine the execution time as the value of minimum support changes from 2% to 6%. The results shown in Figure 3-b demonstrate that the execution time decreases when the number of fuzzy sets increases. So, the execution time is inversely propositional to the minimum support value and the number of fuzzy sets.

In the last experiment in the first set of tests, we examine how the method scales up. For this purpose, we increase the number of transactions from 20K to 100K. The results reported in Figure 4 demonstrate that the proposed method scales well. The mining time required for each of FS3 and FS4 scales quite linearly. However, the execution time of FS3 is larger than that of FS4; this is because the former finds larger number of association rules as shown in Figure 2-b.

5 Summary and Conclusions

The use of the fuzzy set theory in data mining systems when considering quantitative attributes leads to more generalized rules and enhances the understandability of the discovered knowledge. In order to tackle this bottleneck, we proposed in this paper a general architecture that utilizes a fuzzy data cube for knowledge discovery that involves quantitative attributes. We also proposed a mining algorithm for extracting interesting fuzzy rules in a given taxonomy. A top-down progressive deepening technique has been used while mining multi-cross-levels association rules. Also, three important criteria, minimum support, minimum confidence and item importance, used in determining the interestingness of the rules has been represented by fuzzy sets. So, we obtained more

meaningful rules that are more understandable for human beings. As apart from the studies described in the literature, we defined the concept of weighting internal nodes on a given tree. The conducted experiments showed that the proposed approach produces meaningful results and has reasonable efficiency. The results of all the experiments are consistent and encouraging.

References

1. C.C. Agarwal and P.S. Yu, "A new approach to online generation of association rules," *IEEE TKDE*, Vol.13, No.4, pp.527-540, 2001.
2. C.H. Cai, et al, "Mining Association Rules with Weighted Items," *Proc. of IDEAS*, pp.68-77, 1998.
3. M. Delgado, N. Marin, D. Sanchez and M. A. Vila, "Fuzzy Association Rules: General Model and Applications", *IEEE TFS*, Vol.11, No.2, 2003.
4. J. Han, "OLAP Mining: An Integration of OLAP with Data Mining," *Proc. of IFIP ICDS*, pp.1-11, 1997.
5. J. Han, "Towards on-line analytical mining in large databases," *Proc. of ACM SIGMOD*, 1998.
6. J. Han and Y. Fu, "Mining multiple-level association rules in large databases," *IEEE TKDE*, Vol.11, No.5, pp.798-804, 1999.
7. M. Kamber, J. Han and J.Y. Chiang, "Meta-rule guided mining of multidimensional association rules using data cubes," *Proc. of KDD*, pp.207-210, 1997.
8. M. Kaya, R. Alhajj, F. Polat and A. Arslan, "Efficient Automated Mining of Fuzzy Association Rules," *Proc. of DEXA*, Lecture Notes in Computer Science, Springer-Verlag, 2002.
9. M. Kaya and R. Alhajj, "Facilitating Fuzzy Association Rules Mining by Using Multi-Objective Genetic Algorithms for Automated Clustering", *Proc. of IEEE ICDM*, Nov. 2003.
10. M. Kaya and R. Alhajj, "Fuzzy OLAP Association Rules Mining Based Modular Reinforcement Learning Approach for Multiagent Systems," *IEEE TSMC-B*, 2005.
11. M. Kaya and R. Alhajj, "Extending OLAP with Fuzziness for Effective Mining of Fuzzy Multidimensional Weighted Association Rules," *Proc. of ADMA*, Lecture Notes in Computer Science, Springer-Verlag, 2006.
12. C.M. Kuok, A.W. Fu and M.H. Wong, "Mining fuzzy association rules in databases," *SIGMOD Record*, Vol.17, No.1, pp.41-46, 1998.
13. H. Lu, L. Feng and J. Han, "Beyond Intratransaction Association Analysis: Mining Multidimensional Intertransaction Association Rules", *ACM TOIS*, Vol.18, No.4, pp.423-454, 2000.
14. R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," *Proc. of ACM SIGMOD*, pp.1-12, 1996
15. A. K. H. Tung, H. Lu, J. Han and L. Feng, "Efficient Mining of Intertransaction Association Rules", *IEEE TKDE*, Vol.15, No.1, pp. 43-56, 2003.
16. S. Yue, et al., "Mining fuzzy association rules with weighted items," *Proc. of IEEE SMC*, 2000.
17. W. Zhang, "Mining Fuzzy Quantitative Association Rules," *Proc. of IEEE ICTAI*, pp.99-102, 1999.

A Methodological Contribution to Music Sequences Analysis

Daniele P. Radicioni and Marco Botta

Università di Torino, Dipartimento di Informatica
C.so Svizzera 185, 10149, Turin, Italy
{radicion, botta}@di.unito.it

Abstract. In this paper we present a stepwise method for the analysis of musical sequences. The starting point is either a MIDI file or the score of a piece of music. The result is a set of likely themes and motifs. The method relies on a pitch intervals representation of music and an event discovery system that extracts significant and repeated patterns from sequences. We report and discuss the results of a preliminary experimentation, and outline future enhancements.

1 Introduction

In the last few years many efforts have been spent on music issues within the AI community. Two main tasks have been addressed, requiring “intelligent” and sophisticated strategies: music analysis and music performance. The first line of research investigated topics such as performer recognition [1], harmonic analysis [2], segmentation [3], whereas the second one aims at reproducing expressive music performance by means of artificial systems [4,5].

Music analysis is a relevant task, in that it deeply affects our comprehension of music, as regards of composition, performance and listening. Music analysis is a challenging task: consider, e.g., that a significant part of professional music instruction is concerned with assisting the learner in “understanding” music for the different purposes of composition and performance.

In this paper we point out the problem of discovering repeated patterns, as a major issue for building systems for music analysis. It is commonly acknowledged that in Western Tonal Music repetition plays a fundamental role, and individuating themes and motifs is a fundamental step towards discovering higher order blocks, and their dependency relationships as well [6]. For example, looking at a fugue, one would individuate its main constituents (subject and countersubject), and then recognize their disperse episodes, generated through *imitation* and *transposition* [7]. Discovering significant repetitions in music has many applications, such as providing tools for indexing large music corpora and for content-based retrieval from music databases [8]. Moreover, in the area of computer assisted music analysis, both didactic tools and softwares for music performers and analysts would benefit from the discovery of the motifs underlying whole pieces.



Fig. 1. Main theme from *Contrapunctus 4* from J.S. Bach’s *Die Kunst der Fuge*, and the variations obtained through the techniques of inversion, inversion combined with contrary motion, augmentation and diminution

Three main complexity factors increase the difficulty of the task. Meaningful motifs are interleaved with irrelevant gaps. Moreover, we are interested in discovering modified occurrences of motifs, as well. Lastly, we are interested in discriminating motifs from insignificant repetitions, whilst the greater part of repetitions in music are not *perceptually significant* [9]. These issues make the discovery of repeated patterns a challenging problem, and an interesting test-bed for automatic systems.

In this work we propose a novel approach to individuate themes and motifs. We show that the musical patterns discovery can be tackled via a Hierarchical Hidden Markov Model (HHMM) approach: the present system takes as input MIDI files and returns scores, where the patterns found are highlighted, and the corresponding MIDI files of the discovered themes.

The paper is organized as follows. First, we define the problem being solved; then we point out some similarities with other fields where one has to handle episodes represented as strings of symbols. Next, we briefly review the literature, and then introduce our system. We describe the music encoding adopted, and illustrate the system’s basic features. Finally we report about a preliminary experimentation, discussing the obtained results.

2 Patterns in Music

We address the problem of recognizing the most significant motifs, and the problem of recognizing their disperse episodes as well. At all times composers adopted various techniques for composing music from few ideas; but it is from the Renaissance that these techniques were refined to such an extent that it was possible to build entire pieces from a single musical idea. The simpler form of repetition is *literal* repetition, but far more often the repetition is combined with *variation*: changes may involve harmony and/or melody and/or rhythm. Among the most widely used variation techniques, we mention *augmentation* and *diminution* (where the length of the repeated notes is prolonged or shortened), *inversion* of intervals between note pairs and *contrary motion* [10] (Fig. 1). These techniques have been used by both historic and contemporary composers (see, e.g., the works from J.S. Bach (1685–1750) and A. Schoenberg (1874–1951)), in particular

for polyphonic music. Polyphonic music can be defined as a texture consisting of overlapped lines, where each of the individual parts is called *voice*, even if it is played by instruments. Each voice is independent from the other ones (we don't have a melody together with the accompaniment, but rather equally important voices), thus retaining its identity (Fig. 2).

The problem of finding repeated patterns in polyphonic music can be formulated in the following terms. A motif can be thought of as a complex event (CE), occurring sparsely in a sequence of notes. Such a complex event is composed by *episodes* of atomic events (AEs). In turn, episodes are composed by strings of symbols: in our case, by pitch intervals (see below, Section 4.1). In this setting, based on the properties of regular expressions, it is possible to infer a model of the motif being searched via an abstraction mechanism. This approach transports to the musical domain a well established modeling framework, which has been successfully tested on various domains, such as user (typing) profiling [11,12].

3 State of the Art

Several algorithms that tackle the task of pattern recognition in music have been developed. We briefly review the principal and closely related ones, whilst a richer tour is provided in [9]. Dovey [13] proposes an algorithm for querying musical databases, where music events are represented as strings of (sets of) note pitches. Different kinds of relationships can be individuated between events, such as about harmony, or according to whether the two events under consideration fall into some pitch range. This approach also requires to specify the dimension of gaps between consecutive events. The pattern discovery technique devised by Conklin & Anagnostopoulou [14] relies on a representation based on a set of parameters (*viewpoints*, in the authors terms [15]): here each string represents an individual parameter, and meaningful repetitions are individuated based on a Hidden Markov Model, looking for repetitions more frequent than expected. These approaches only discover *exactly* repeated factors in strings.

On the other side, the algorithm by Rolland [16], which allows retaining much musical information such as duration, interval, degree in the overall tonality, has been designed to discover *approximately* repeated patterns. The author defines a similarity metrics between pairs of sequence segments: the Multi-Description Valued Edit Model implements a function for computing the *edit distance* between strings [17]. This algorithm allows discarding as not similar all the patterns below some threshold k , thus allowing to define a notion of *k-similarity*. Then, the distance between each pattern instantiation and the pattern itself is computed. Based on this proximity score, Rolland obtains a list of the most prominent repeated patterns. Cambouropoulos et al. [18] propose an algorithm for approximate string matching, by transporting to the music domain the well-known concept of *edit distance*.

Lartillot [19] devised an interesting mechanism for pattern induction, based on a plain encoding of pitch intervals and metrical salience. The algorithm first



Fig. 2. Bars 1–7 from Fugue in C major from Book 1 of J.S. Bach’s *Das Wohltemperierte Klavier*: each voice states the subject

analyses note pairs within a temporal window of fixed size. At this step similar intervals are retained as potential patterns. Then, Lartillot’s algorithm checks the prosecution of repeated patterns by considering only *melodic contour* (sequences being considered can prosecute upward, downward, or constant). This way, individuating approximate repetitions is no longer an issue. Also, this system is provided with a sort of abstraction mechanism, defining pattern *occurrences* as instances of pattern *classes*. However, a limitation of the approach resides in the fact that the algorithm cannot handle long sequences, due to the high computational cost necessary for this kind of analysis.

One popular approach for discovering repeated patterns is the “geometric” approach by Meredith et al. [9]: music is represented as a multidimensional dataset. Each event in the score can be encoded via an arbitrary number of dimensions, such as onset time, pitch, duration and voice. The authors define perceptually relevant repetitions in terms of the maximum pattern that can be translated into another pattern in the dataset. This is equivalent to finding out all the transposition-invariant occurrences of a pattern, according to a given dimension/set of dimensions. Unlike the above mentioned works, this approach allows handling polyphonic music.

4 The System

4.1 Encoding Musical Information

We take as input pieces encoded as standard MIDI files¹, where each musical “voice” is represented as a separate track: given n voices, we have as many tracks. For example, the piece in Fig. 2 can be encoded with 4 tracks (Fig. 3). We then extract from each track a sequence of music events: an event has a pitch, onset and offset; each new onset determines a new event.

Underpinned by music analysis [7] and music cognition [6] literatures, we are primarily concerned with music *intervallic* content, in that it can reveal pitch contour commonalities between motifs. For example, let us consider the excerpt presented in Fig. 2. Here we have a “curve” shaped motif (properly, the

¹ <http://www.midi.org/>

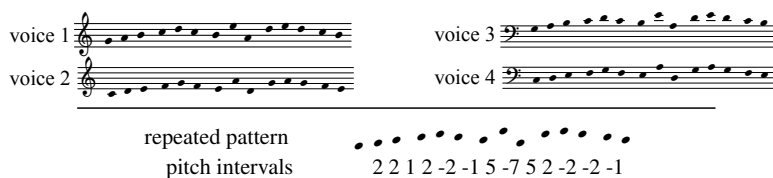


Fig. 3. Top: the subject of the fugue presented in Fig. 2 is extracted from each voice. Bottom: all the voices exhibit the same shape, despite the notes are actually different. Commonalities among the different occurrences of the subject can be seen via a pitch interval representation.

subject of the fugue), that accomplishes an ascending–jump–descending melodic movement. The essence of this kind of motifs –see e.g., the top of Fig. 3– is best grasped if one considers the pitch intervals between each pair of notes, as it is shown in the bottom of Fig. 3. We can see that the four occurrences of the subject are invariant under transposition, so that the pattern can be individuated considering the intervals rather than the actual notes. More sophisticated kinds of representation have been designed, such as Conklin’s *multiple viewpoints* [15], that in principle could better fit to pattern discovery. However, as it was pointed out by Conklin [14], a systematic experimentation over 185 J.S. Bach chorales provided evidence that patterns were mainly found via “melodic intervals”. As a consequence, we chose a simple representation accounting for intervals only; this has the advantage of permitting to translate music input into a string with a smaller alphabet.

4.2 Extracting and Modeling Motifs

In order to accomplish this step, we used a recently proposed event discovery system [11,12]. The main idea is that of modeling each motif by means of a Profile Hidden Markov model (PHMM), and representing a sequence of motifs interleaved with gaps by a Hierarchical Hidden Markov model (HHMM).

A Hidden Markov Model (HMM) is a stochastic finite state automaton [20] defined by a tuple $\lambda = \langle Q, O, A, B, \pi \rangle$, where:

- Q is a set of states, and O is a set of atomic events (observations),
- A is a probability distribution governing the transitions from one state to another. Specifically, any member $a_{i,j}$ of A defines the probability of the transition from state q_i to state q_j , given q_i .
- B is a probability distribution governing the emission of observable events depending on the state. Specifically, an item $b_{i,j}$ belonging to B defines the probability of producing event O_j when the automaton is in state q_i .
- π is a distribution on Q defining, for every $q_i \in Q$, the probability that q_i is the initial state of the automaton.

The difficulty, in this basic formulation, is that, when the set of states Q grows large, the number of parameters to estimate (A and B) rapidly becomes intractable. Nevertheless, the size of the parameters to estimate can be strongly reduced if one defines a structure for the automaton, where a number of state transitions and the possible emissions in a state are cut a priori. This corresponds to setting to 0 the corresponding values in matrix A and B . Actually, many work related to HMM applications consists in handcrafting the a priori structure of the automaton, as, for instance, the Profile Hidden Markov Model [21], widely used for DNA analysis.

A Profile Hidden Markov Model assumes that a motif has a canonical instantiation form, that can be affected by insertion and deletion errors. A PHMM is basically a forward graph with three categories of states:

- *match* states, where the emission corresponds to the unique symbol expected in the canonical instantiation;
- *insertion* states, where a number of symbols due to random noise can be inserted;
- *deletion* states, where non emission occurs where it was supposed to.

In addition, other two non emitting states are required: the *start* state, and the *end* state. Recursion is considered only in the form of self-loops associated to *insertion* states. It has been experimentally shown that PHMM is more accurate than string matching to detect motifs [21], and then we expect that it holds for musical sequences as well. Moreover, the algorithmic complexity inherent to PHMM is linear both for Viterbi and Forward-Backward algorithms.

The Hierarchical HMM (HHMM) proposed by Fine, Singer and Tishby [22] is an extension of the basic HMM, that immediately follows from the regular languages property of being closed under substitution; this property allows a large finite state automaton to be transformed into a hierarchy of simpler ones. More specifically, a HHMM is a hierarchy where, by numbering the hierarchy levels with ordinals increasing from the lowest towards the highest level², observations generated in a state q_i^k by a stochastic automaton at level k are sequences generated by an automaton at level $k - 1$. The emissions at the lowest levels are again single tokens as in the basic HMM. Moreover, no direct transition may occur between the states of different automata in the hierarchy. As in HMM, in every automaton the transitions from state to state is governed by a distribution A and the probability for a state being the initial state is governed by a distribution π . The restriction is that there is only one state which can be the terminal state.

The major advantage provided by the hierarchical structure is a strong reduction in the number of parameters to estimate. In fact, automata at the same level in the hierarchy do not share interconnections: every interaction through them is governed by transitions at the higher levels.

A second advantage is that, as it will be described in the following, the modularization enforced by the hierarchical structure allows the different automata

² We use a reverse numeration for hierarchy levels, with respect to the original formulation in [22].

to be modified and trained individually, thus providing a natural subproblem decomposition.

The basic learning algorithm, fully described in [11,12] builds the HHMM hierarchy bottom-up starting from the lowest level (actually, only two levels are built). The first step consists in searching for possible motifs, i.e., short chains of consecutive symbols that appear frequently in the learning traces, by means of classical methods used in DNA analysis [21]. A PHMM is then built from the found motifs. As models of the motifs are constructed independently from one another, it may happen that models for spurious motifs are constructed. At the same time, it may happen that relevant motifs are disregarded just because their frequency is not high enough. Both kinds of problems will be fixed at a later time. Starting from the models found at this step, a HHMM can be built in the following way: the input sequences are abstracted (i.e., rewritten in terms of the models found) by substituting each occurrence of a PHMM with a symbol in a new alphabet and subsequences between two motifs not attributed to any PHMM, by means of a special symbol called *gap*.

After this basic cycle has been completed, an analogous learning procedure is repeated on the abstracted sequences, where models are now built for sequences of *episodes*, searching for co-occurrent motifs. In this process, spurious motifs not showing significant regularities can be discarded. The major difference with respect to the first step, is that models built from the abstracted sequences are observable Markov models. This makes the learning task easier and decreases its computational complexity. After building the HHMM structure in this way, it can be refined by using standard training algorithms [23,22].

4.3 Producing the Results

Once the HHMM has been built, we are ready to produce the final results of the system. In this last step, the acquired model is instantiated on every voice and two outputs are generated: a score of the piece tagged with the motifs found in that voice. In order to provide also audible results, we generate as many MIDI files as the found motifs (i.e., for every motif we produce a MIDI file that contains the notes in the motif). Both scores and MIDI files are available on the Web³.

5 Experimental Validation

In order to assess the results provided by the system, we have collected a dataset composed by the fugues from Book 1 of J.S. Bach's *Das Wohltemperierte Klavier*. We used MIDI recordings retrieved from BachCentral.com⁴. Due to problems encountered while processing the MIDI files⁵, we could actually use only 15 of the 24 fugues from the original corpus.

³ <http://www.di.unito.it/~botta/bach-fugues>

⁴ <http://www.bachcentral.com/wtcMidi1.html>

⁵ For reading the files and extracting the tracks, we used the parsing routines provided by the standard `javax.sound.midi` package.

We have run the system to discover the main motif (i.e., the subject) of each fugue. Also, we have been looking for all the occurrences of such subject throughout the fugue: specifically, we have been looking for the occurrences of the whole (if varied) subject, thus disregarding episodes such as the *stretti*, where only the head or the tail of the subject occurs. While testing the system on fugues motifs only, we considered motifs longer than a fixed amount (presently, this was set to 6 notes), thereby retaining motifs that have a score higher than the average of the scores left. The experimentation should answer the following questions: *i*) How much do such filtered data fit to the actual subjects; *ii*) How many of the subject repetitions do we find; *iii*) How many notes in the (repeated) subjects are actually individuated.

i) The system correctly recognized the subject in 72.55% of cases. Overall *ii*), 74.73% of the total number of subject occurrences were identified; also, on average, *iii*) we have individuated 80.86% of the notes in the subjects.

5.1 Discussion

The present results hardly compare with literature, because, to the best of our knowledge, no researchers addressing the pattern discovery in the musical domain have tested their systems in a systematic way. Rather, examples of individual patterns have been provided, or the most discriminative musical features have been pointed out –e.g., pitch intervals, contour– (see, respectively, the state of art systems presented in [9] and [14]). Reasonably, providing crude numbers would result improper in some cases, but perhaps the lack of systematic experiments witnesses about the difficulty of the task, as well. Hence our results can furnish a baseline against which other systems can be compared.

Provided that this preliminary experimentation considered only a small dataset, and that we have performed only a reduced form of automatic analysis, aiming at discovering vividly *individuated* motifs, the results are satisfying.

A closer look to the data may be helpful in completing the assessment of the results, and in suggesting some criteria for future improvements. In the test *i*) we identify the subject in 72.55% of cases, but if we retain all the results without filtering, then the success ratio raises to 100% of cases. Therefore, taken for granted that models for subjects can be acquired, refinements to the filtering function are at hand. For what concerns test *ii*), some improvements would be possible by considering additional music information, such as, e.g., the *absolute* intervals. Most likely this would be useful to discover occurrences of the subject with *inverted* (Fig. 1) intervals. Grasping these cases (Fig. 4) would further improve the system’s accuracy. The test *iii*) would benefit from a slightly different splitting of the string in input.

However, the results also show that much work has still to be done. Many errors were committed when encountering “too short” subjects (e.g., Fugue 4), where it is harder to acquire a significant pattern. Moreover, we were penalized from cases where “real answer” significantly transforms the expositions of the subject (e.g., Fugue 21), thus increasing the difficulty to individuate the common underlying model.



Fig. 4. Top: the subject of Fugue 20; bottom: the subject is stated by the *soprano* voice with the inversion of intervals

6 Conclusions

This paper has presented a methodology for addressing a captivating problem: analyzing the *horizontal* textures of music. Namely, we have applied it for discovering the subjects in some fugues from J.S. Bach's *Das Wohltemperierte Klavier*. A working system implements the methodology: based on a plain though effective encoding of music, it acquires a model of the most significant repeated patterns. Then it outputs the results both as analyzed scores and as MIDI files, containing the patterns discovered. The experimental validation demonstrated the adequacy of the approach, while also pointing out some open issues, that will steer our future work.

Furthermore, the hierarchical hidden Markov model learned by the system could provide useful insights about the overall structure of musical pieces, allowing to describe them in terms of basic building blocks (motifs and gaps) and their relationships.

Acknowledgments

This work has been partly funded by the University of Piemonte Orientale that supported Daniele Radicioni. We would also thank Attilio Giordana and Ugo Galassi for working to the core system.

References

1. Stamatatos, E., Widmer, G.: Automatic identification of music performers with learning ensembles. *Artificial Intelligence* **165** (2005) 37–56
2. Conklin, D.: Representation and discovery of vertical patterns in music. In Anagnostopoulou, C., Ferrand, M., Smaill, A., eds.: *Music and Artificial Intelligence: Procs. ICMAI*. Volume 2445 of *LNAI.*, Springer-Verlag (2002) 32–42
3. Bod, R.: A Unified Model of Structural Organization in Language and Music. *Journal of Artificial Intelligence Research* **17** (2002) 289–308
4. Widmer, G.: Modeling the Rational Basis of Musical Expression. *Computer Music Journal* **19**(2) (1995) 76–96
5. Lopez de Mantaras, R., Arcos, J.: AI and Music: From Composition to Expressive Performance. *AI Magazine* **23** (2002) 43–57

6. Lerdahl, F., Jackendoff, R.: *A Generative Theory of Tonal Music*. MIT Press, Cambridge, MA (1983)
7. Bent, I., ed.: *Music Analysis in the Nineteenth Century*. Cambridge University Press, Cambridge (1994)
8. Hsu, J.L., Liu, C.C., Chen, A.: Efficient Repeating Pattern Finding in Music Databases. In: *Procs. of International Conference on Knowledge and Information Management*. (1998) 281–288
9. Meredith, D., Lemström, K., Wiggins, G.: Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research* **31**(4) (2002) 321–245
10. de la Motte, D.: *Kontrapunkt. Ein Lese und Arbeitsbuch*. Kassel, Bärenreiter (1981)
11. Botta, M., Galassi, U., Giordana, A.: Learning Complex and Sparse Events in Long Sequences. In: *Procs. of the 16th European Conference on Artificial Intelligence, Valencia, Spain* (2004) 425–429
12. Galassi, U., Giordana, A., Saitta, L., Botta, M.: Learning Regular Expressions from Noisy Sequences. In: *Procs. of 6th International Symposium on Abstraction, Reformulation and Approximation, Edinburgh, Scotland (UK)* (2005) 92–107
13. Dovey, M.: A Technique for “regular expression” style searching in polyphonic music. In: *Procs. of the Internat. Music Information Retrieval Conference*. (2001)
14. Conklin, D., Anagnostopoulou, C.: Representation and Discovery of Multiple Viewpoint Patterns. In: *Procs. of the 2001 International Computer Music Conference*. (2001) 479–485
15. Conklin, D., Witten, I.H.: Multiple viewpoint systems for music prediction. *Journal of New Music Research* **24**(1) (1995) 51–73
16. Rolland, P.Y.: Discovering Patterns in Musical Sequences. *Journal of New Music Research* **28**(4) (1999) 334–350
17. Navarro, G.: A Guided Tour to Approximate String Matching. *ACM Computing Surveys* **33**(1) (2001) 31–88
18. Cambouropoulos, E., Crochemore, M., Iliopoulos, C., Mouchard, L., Pinzon, Y.: Algorithms for Computing Approximate Repetitions in Musical Sequences. *Intern. J. Computer Math.* **79**(11) (2002) 1135–1148
19. Lartillot, O.: Discovering Musical Patterns through Perceptive Heuristics. In: *Procs. of International Symposium on Music Information Retrieval*. (2003)
20. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of IEEE* **77**(2) (1989) 257–286
21. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: *Biological sequence analysis*. Cambridge University Press (1998)
22. Fine, S., Singer, Y., Tishby, N.: The hierarchical hidden markov model: Analysis and applications. *Machine Learning* **32** (1998) 41–62
23. Murphy, K., Paskin, M.: Linear time inference in hierarchical hmms. In: *Advances in Neural Information Processing Systems (NIPS-01)*. Volume 14. (2001)

Towards a Framework for Inductive Querying^{*}

Jeroen S. de Bruin

Leiden Institute of Advanced Computer Science
Leiden University, The Netherlands
jdebruin@liacs.nl

Abstract. Despite many recent developments, there are still a number of central issues in inductive databases that need more research. In this paper we address two of them. The first issue is about how the discovery of patterns can use existing patterns. We will give a concrete example showing an advantage of mining both the patterns and the data. The second issue we consider is the actual implementation of inductive databases. We will propose an architectural framework for inductive databases and show how existing databases can be incorporated.

1 Introduction

In Knowledge Discovery in Databases (KDD) the data volumes are becoming increasingly larger and efficient KDD becomes an important issue. In order to cope with the large volumes of data and patterns, Imielinsky and Mannila proposed the concept of so-called inductive databases [5].

An inductive database, as defined in [4], is a database that stores data as well as patterns as first class objects. More formally, an inductive database $IDB(D, P)$ has a data component D and a pattern component P . Storing patterns as first-class citizens in a database enables the user to query them in a same way as the data can be queried. The extra power lies in the so-called *crossover* queries which contain both pattern and data elements.

Despite many recent developments, there are still a number of central issues in inductive databases that need more research. In this paper we address two of them. First, the question: "if we have data and patterns in our inductive database, how does this help us to discover other patterns?". We will give a concrete example showing the advantage of mining both the patterns and the data. The second issue we consider is the actual implementation of inductive databases. We will propose a method for combining existing patternbases and databases into an inductive database.

The paper is organized as follows. In Section 2 we give concrete examples that show that it can be useful to combine data and patterns. Section 3 proposes a software architecture for inductive databases. Section 4 concludes and suggests further research.

^{*} This research is carried out within the Netherlands Bio Informatics Consortium (NBIC) BioRange Project.

2 Example

In this section we will show through some concrete examples how in an inductive database the existence of patterns can speed up a data mining query. Consider the following problem.

Suppose we have a data set in which the entries are labeled with a class. We want to find all item sets that correlate with classes.

We use the χ^2 statistic, that is we are interested in all patterns (in our case, the patterns are items sets, which are combinations of attribute values that are shared by a subset of all the records in the data) that have a χ^2 value above a certain threshold (the threshold is to be fixed in advance). The χ^2 value is computed as follows. Assume that there are d classes and, given an item set, construct a vector $\mathbf{a}=(a_1, \dots, a_d)$ of fractions a_i , where a_i is the fraction of examples in class i that contain the item set. Then

$$\chi^2(\mathbf{a}) = \sum_{i=1}^d \frac{(O_{i1} - E_{i1})^2}{E_{i1}} + \frac{(O_{i2} - E_{i2})^2}{E_{i2}}.$$

where

- $E_{i1} = (\sum_{i=1}^d (a_i n_i) n_i) / N$: the expected number of elements in class i that contain the pattern,
- $E_{i2} = (\sum_{i=1}^d ((1 - a_i) n_i) n_i) / N$: the expected number of elements in class i that do not contain the pattern,
- $O_{i1} = a_i n_i$: the observed number of elements in class i that contain the pattern,
- $O_{i2} = (1 - a_i) n_i$: the observed number of elements in class i that do not contain the pattern

where n_i is the number of examples in class i and $N = \sum_{i=1}^d n_i$ the total number of examples.

We are going to compare two ways to generate the item sets with a χ^2 above a threshold: a direct method and a method that prunes the search space using other patterns.

In the *direct method* we generate every possible subset of the data. Suppose we have a table of data which contains m attributes, and for each attribute i , $1 \leq i \leq m$, where a_i is the number of possible values that attribute can have. Then the number of possible subsets s (excluding the empty set) is:

$$s = \left(\prod_{i=1}^m (a_i + 1) \right) - 1.$$

Since s is the number of subsets, this is also the number of times that χ^2 must be evaluated.

For the *pruning method* we use an idea of [8]. In that paper it is proposed to check only frequent item sets (above a class dependent threshold) within the

classes. Given a χ^2 threshold θ a minimum frequency threshold θ_i for class i is derived:

$$\theta_i = \frac{\theta N}{N^2 - n_i N + \theta n_i}.$$

For for an item set to a χ^2 larger than θ , its frequency a_i in at least one of the classes i should be large than θ_i :

$$a_1 n_1 \geq \theta_1, \dots, a_d n_d \geq \theta_d.$$

A χ^2 threshold is often chosen by picking a p-value, from which the threshold value θ can be derived. These frequency thresholds already present two options for optimization. If the frequent item sets are already present in the inductive database, then we can use the frequency threshold to prune all patterns that fall below the calculated frequencies, thereby lowering the number of times the χ^2 measure has to be calculated. If not all frequent item sets are known, then the θ_i can be used as a parameter for a frequent mining set algorithm. In our experiments we want to find out exactly how much calculations of the χ^2 measure are saved when the patterns are already present.

The data sets used in the experiments were obtained from the UCI [7]. More specific information about the data used are in the table below. For the generation of frequent item sets we used the implementation of Borgelt [3].

Description	Size(N)	Classes(d)	
Mushrooms	8124	2	Classes: e(4208) and p(3916)
Chess KRK	28056	18	Class max: 4553, class min: 27

First consider the mushroom database. The data-scheme contains 22 different attributes, all ranging from 2 to 12 different values per attribute. The direct method checks $1.6346E17$ subsets, and thus has to do that many χ^2 evaluations.

Now consider the second method. We first split the database into two data sets, each containing the data of one class, and then look for patterns that are significant according to the χ^2 measure. We do this for both data sets and we merge the resulting pattern sets. The resulting number of patterns is the maximum number of patterns that have to be evaluated with the χ^2 statistic. (There might be patterns that are in both sets.) The results of the experiments with a number of p-values are in the table below, along with the reduction ratio in the number of χ^2 evaluations.

p	#gen. patterns	#subsets	ratio
10^{-3}	525310	$1.6346E17$	$3.21371E-12$
10^{-5}	466686	$1.6346E17$	$2.85506E-12$
10^{-8}	408894	$1.6346E17$	$2.50150E-12$
10^{-10}	399870	$1.6346E17$	$2.44630E-12$

As can be seen, the reduction in χ^2 evaluations is large. Of course the number of evaluations depends on the number of classes, the number of attributes and the number of values each attribute can take and especially the last two are quite large here. So let us analyze this method's efficiency when the number of attributes and attribute values are lower. Consider the King-Rook-King data set. This data-scheme contains only six attributes, each consisting of eight possible values. For the direct method this yields a total of 531441 subsets. The ratios are given below.

p	#gen. patterns	#subsets	ratio
10^{-3}	72990	531441	0.1373
10^{-5}	61996	531441	0.1167
10^{-8}	58018	531441	0.1092
10^{-10}	56388	531441	0.1061

As can be seen the ratios are much higher when less attributes and attribute values are involved, but the reduction in χ^2 calculations is still a little below 90%. Note that more optimizations are possible, both in the number of evaluations as in the evaluations themselves. In the results above the number of generated pattern were the sum of all generated patterns per class, but we did not check if patterns occurred more than once.

Another optimization is the evaluation of χ^2 itself. When generating patterns with the apriori-algorithm, the coverage, which is also used in the O_{i1} and E_{i1} of the χ^2 measure, is stored in the pattern and thus need not to be calculated again, speeding up the calculation of χ^2 . To be fair, it should be stated that there are also many ways to optimize the χ^2 calculations for the direct method.

Concluding, our examples show that it can be advantageous to use existing patterns to derive new patterns. We are convinced that there are many more similar situations in which existing patterns are useful.

3 Software Architecture for an Inductive Database

Next we come to the implementation of inductive databases. In this section we will propose a software architecture. The rationale behind a software architecture for inductive databases is clear; by creating software architectures software becomes better, lasts longer and contains fewer errors. However, although much research has been done on various aspects of inductive databases, the implementation of an inductive database has received very little attention, but is still vital for performance issues, and extensibility of the database system.

Before we discuss the software architecture, we first want to address that the distinction between patterns and data is not only an intuitive one: the patterns and the data differ in a number of aspects. While raw data usually has a rigid structure, patterns usually lack this structure. Indeed, studies in the PANDA project (<http://dke.cti.gr/panda/>) have shown that storing patterns in

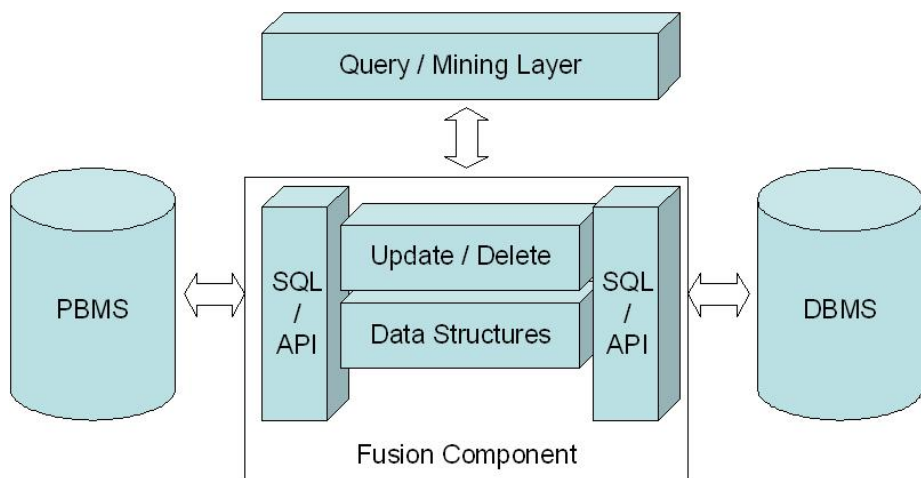


Fig. 1. Fusion architecture

a relational way can be very inefficient, due to their semi-structured nature [1]. *Therefore, we propose that an inductive database has a separate database and a separate patternbase, connected by a fusion component as outline in Figure 1.*

We first discuss the patternbase and then the fusion component of our Fusion architecture.

According to [2], in a patternbase a PBMS (Pattern Base Management System) should contain three layers: a pattern layer containing the patterns, a pattern type layer containing the pattern types, and a class layer that contains pattern classes: collections of semantically related patterns.

Regardless of how a pattern is represented within the patternbase, a pattern has at least the following information attached to it:

- The pattern source s , i.e. the table(s) or the view from which the pattern is derived.
- The pattern function f , which is the procedure used to acquire the pattern. This function is situated in the Query / Mining layer.
- The pattern parameter collection P , which is a (possibly empty) list of parameter values used by the procedure.

We need this information in order to update patterns in case their source tables change.

Current relational databases are unfit to represent such an architecture and XML databases have been proposed to store and represent patterns [6,1]. *We also propose to use an XML database for the patternbase.*

The fusion component is responsible for database communication and synchronization of patterns with their corresponding source data, and for maintaining data structures that allow these procedures to proceed as efficiently as possible. Communication to the database and patternbase proceeds either through SQL statements or through a database API, if present.

The fusion component implements the following pattern-data synchronization operations:

- *Recalc*(r), which recalculates the patterns in the patternbase affected by a change of database relation r , according to specified function f and parameter values P over source s .
- *Del*(r), which deletes a pattern if (part of) its source s is no longer present in the database.

4 Conclusions and Future Work

In this paper we focussed on two aspects of inductive databases that have received relatively less attention in that field. We gave a concrete example that showed that patterns can speed up the KDD process. We also proposed a software architecture for the construction of an inductive database through the extension of a database with a patternbase.

We need to extensively test our architecture with various data in order to ensure it is maximized for extensibility and efficiency, two traits that can become contradicting in an architecture. Furthermore, still much research needs to be done on pattern representation and inductive query languages in areas other than frequent pattern mining.

References

1. E. Bertino, B. Catania, E. Kotsifakos, A. Maddalena, I. Ntoutsis, Y. Theodoridis. PBMS Querying and Storage Issues, PANDA Technical Report PANDA-TR-2004-02, Feb. 2004.
2. E. Bertino, B. Catania, and A. Maddalena. Towards a Language for Pattern Manipulation. In *Proc. 1st International Workshop on Pattern Representation and Management (PaRMA04)*, Greece, March 2004.
3. C. Borgelt. Efficient Implementations of Apriori and Eclat. In *Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL)*. CEUR Workshop Proceedings 90, Aachen, Germany 2003. <http://www.ceur-ws.org/Vol-90/>
4. L. De Raedt. A perspective on inductive databases. In *SIGKDD Explorations*, 4:69-77, 2003.
5. T. Imielinski and H. Manilla. A Database Perspective on Knowledge Discovery. In *Communications of the ACM*, 39(11):58-64, 1996.
6. R. Meo, G. Psaila. Toward XML-Based Knowledge Discovery Systems. In *Proc. of the IEEE International Conference on Data Mining*, pp. 665-668, Japan, 2002.
7. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J. (1998). UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mlearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science.
8. S. Nijssen and J.N. Kok. Multi-Class Correlated Pattern Mining. In *Proceedings of the Fourth International Workshop on Knowledge Discovery in Inductive Databases*, Porto, LNCS 3933, 2006.

Towards Constrained Co-clustering in Ordered 0/1 Data Sets

Ruggero G. Pensa, Céline Robardet, and Jean-François Boulicaut

INSA Lyon, LIRIS CNRS UMR 5205

Bâtiment Blaise Pascal

F-69621 Villeurbanne cedex, France

{Ruggero.Pensa, Celine.Robardet, jfboulicaut}@insa-lyon.fr

Abstract. Within 0/1 data, co-clustering provides a collection of bi-clusters, i.e., linked clusters for both objects and Boolean properties. Beside the classical need for grouping quality optimization, one can also use user-defined constraints to capture subjective interestingness aspects and thus to improve bi-cluster relevancy. We consider the case of 0/1 data where at least one dimension is ordered, e.g., objects denotes time points, and we introduce co-clustering constrained by interval constraints. Exploiting such constraints during the intrinsically heuristic clustering process is challenging. We propose one major step in this direction where bi-clusters are computed from collections of local patterns. We provide an experimental validation on two temporal gene expression data sets.

1 Introduction

Many data mining techniques have been designed to support knowledge discovery from 0/1 data, i.e., Boolean matrices whose the rows denote objects and the columns denote Boolean attributes recording object properties.

Table 1. A Boolean context $\mathbf{r}$

	g_1	g_2	g_3	g_4	g_5
t_1	1	0	1	1	0
t_2	0	1	0	0	1
t_3	1	0	1	1	0
t_4	0	0	1	1	0
t_5	1	1	0	0	1
t_6	0	1	0	0	1
t_7	0	0	0	0	1

For instance, given $\mathbf{r}$ in Table 1, object t_1 satisfies only properties g_1 and g_3 , and g_4 . Exploratory data analysis processes often make use of clustering techniques to get insights about global patterns within the data, i.e., to propose partitions of objects and/or of properties such that a grouping quality measure is

optimized. Many efficient algorithms can provide good partitions but suffer from the lack of explicit cluster characterization. This has motivated the research on conceptual clustering, e.g., the co-clustering approaches [1,2,3,4]. Co-clustering goal is to compute bi-clusters, i.e., associations of (possibly overlapping) sets of objects with sets of properties. An example of an interesting bi-partition in $\mathbf{r}$ is $\{\{\{t_1, t_3, t_4\}, \{g_1, g_3, g_4\}\}, \{\{t_2, t_5, t_6, t_7\}, \{g_2, g_5\}\}\}$. The first bi-cluster indicates that the characterization of objects from $\{t_1, t_3, t_4\}$ is that they almost always share properties from $\{g_1, g_3, g_4\}$. Also, properties in $\{g_2, g_5\}$ are characteristic of objects in $\{t_2, t_5, t_6, t_7\}$.

Given a clustering algorithm, the analyst has generally a weak control on the clusters he/she obtains. Typically, he/she can decide for ad-hoc parameter settings which are quite operational and conceptually far from the declarative specification of desired properties. A co-clustering algorithm tries to optimize an objective function (e.g., Goodman-Kruskal's τ coefficient in [1] or the loss of mutual information in [2]) but it might also ensure that some user-defined constraints are satisfied (e.g., the fact that some objects and/or properties have to be together or not). In other terms, we would like to support the search for relevant bi-clusters by enabling user-defined selection predicates (i.e., the conjunction of the objective function optimization constraint with the other user-defined constraints) on bi-partitions as if every possible bi-partition had been computed beforehand. We all know that such a computation is not possible in practice. It explains while a typical (co-)clustering algorithm like COCLUSTER [2] uses heuristic local optimization to provide a good bi-partition without being able to guarantee the optimal one (i.e., the optimization constraint is relaxed). It explains also that using other user-defined constraints is challenging: enforcing some constraints might lead to lower values for the objective functions. Indeed, to the best of our knowledge, only preliminary approaches have concerned mono-dimensional constrained clustering for simple types of user-defined constraints, mainly the so-called must-link and cannot-link constraints [5,6,7,8].

In this paper, we address the problem of (bi-)cluster discovery when at least one of the dimensions is ordered and when interval constraints are defined w.r.t. orders. One of the typical application domains which motivates our study is temporal gene expression data analysis. In this case, objects denotes gene expression level measurements performed for successive time points on a given organism (e.g., major phases of the developmental cycle), and properties are Boolean gene expression properties (e.g., gene up-regulation or over-expression). For a given organism and during its live cycle, groups of genes are activated and then inhibited, being somehow characteristic of some development stages. A biologist might be interested in finding such co-regulated genes to putatively assign some biological functions and (co-)clustering is a popular techniques for this [4]. In our experience, it is however possible that known stages of a development cycle are not really identified by available clustering algorithms. Bi-clusters may contain samples from different stages, or involve time points which are not contiguous. This is quite confusing for biological interpretation. On another hand, the temporal relationship between the sets of biological conditions can be so strong

that all clustering algorithms return perfect time intervals. In that case, when studying interactions between genes which are co-regulated in different stages, it would be nice to enforce the algorithm to look at alternative bi-partitions, i.e., with no mapping to the development stages.

Therefore, our contribution is twofold. First, we consider constraint-based co-clustering on ordered data and this gives rise to new types of constraints. It is then possible to specify whether a collection of bi-clusters has to be consistent w.r.t. these orders, i.e., the so-called *Interval* and *Non-interval* constraints. It enables to get clusters associated to time intervals or to space regions. Our second contribution concerns the framework which is used to compute the bi-partitions given the specified constraints. We recently proposed a generic framework to compute bi-clusters based on collections of local patterns which capture locally strong associations [9]. From an algorithmic point of view, we show here that it is possible to extend it towards constraint-based bi-cluster mining. A related work in gene expression data analysis is [10]. It provides an algorithm to compute clusters which are constrained by some local patterns maximizing the interclass variance. In such a proposal, some local patterns are used to constrain the partition but they are not selected w.r.t. a declarative specification. Our goal is indeed to build a bi-partition which satisfies user-defined declarative constraints via a preliminary selection of local patterns.

Section 2 provides the problem setting, including the definition of bi-cluster constraints. Section 3 recalls the framework from [9] and discusses its extension towards constrained-based co-clustering. Section 4 concerns our experimental validation on real gene expression data sets. Section 5 concludes.

2 Problem Setting

Assume a set of objects $\mathcal{T} = \{t_1, \dots, t_m\}$ and a set of Boolean properties $\mathcal{G} = \{g_1, \dots, g_n\}$. The Boolean context to be mined is $\mathbf{r} \subseteq \mathcal{T} \times \mathcal{G}$, where $r_{ij} = 1$ if property g_j is satisfied by object t_i . For the sake of clarity, $\mathcal{D}$ denotes either $\mathcal{T}$ or $\mathcal{G}$. We define the co-clustering task as follows: we want to compute a partition $P^{\mathcal{T}}$ of K clusters of objects (say $\{P_1^{\mathcal{T}}, \dots, P_K^{\mathcal{T}}\}$) and a partition $P^{\mathcal{G}}$ of K clusters of properties (say $\{P_1^{\mathcal{G}}, \dots, P_K^{\mathcal{G}}\}$) with a bijective mapping denoted σ between both partitions s.t. each cluster of objects is characterized by a single cluster of properties ($\sigma : P^{\mathcal{T}} \rightarrow P^{\mathcal{G}}$). The computed bi-partition is denoted $\mathcal{P} = \{P_1, \dots, P_K\}$ s.t. $P_i = (P_i^{\mathcal{T}}, \sigma(P_i^{\mathcal{T}}))$. Assume now that a real value $s(x_i)$ is associated to each element $x_i \in \mathcal{D}$, where function $s : \mathcal{D} \rightarrow \mathbb{R}$. For instance, $s(x_i)$ can be a temporal or spatial measure related to x_i . In microarray data, where $\mathcal{T}$ is a set of DNA chips, and $\mathcal{G}$ is a set of genes, $s(t_i)$ might be the sampling time related to the DNA chip t_i . On another hand, $s(g_i)$ might measure the absolute position in the whole DNA sequence (if known). The function s , allows to define an order $\preceq$ on dimension $\mathcal{D}$. We say that $x_i \preceq x_j$ iff $s(x_i) \leq s(x_j)$. For the sake of simplicity, if a function s exists on dimension $\mathcal{D}$, then all its elements x_i are ordered, i.e., $\forall i, j$ s.t. $i < j$, $s(x_i) \leq s(x_j)$. We can now redefine the co-clustering task by taking into account the information about the ordered dimension.

Definition 1 (interval and non-interval constraint). *If an order $(\preceq)$ is defined on $\mathcal{D}$, an interval constraint on this dimension, denoted $\mathcal{C}_{int}(\mathcal{D}, \mathcal{P})$, enforces each cluster on $\mathcal{D}$ to be an interval: $\forall k = 1 \dots K$, if $x_i, x_j \in P_k^{\mathcal{D}}$ then $\forall x_l$ s.t. $x_i \preceq x_l \preceq x_j$, $x_l \in P_k^{\mathcal{D}}$. A non-interval constraint denoted $\mathcal{C}_{non-int}(\mathcal{D}, \mathcal{P})$ specifies that clusters on $\mathcal{D}$ should not be intervals: $\forall k = 1 \dots K$, $\exists x_i, x_j \in P_k^{\mathcal{D}}$, $\exists x_l \in \mathcal{D}$ s.t. $x_i \preceq x_l \preceq x_j$, $x_l \notin P_k^{\mathcal{D}}$.*

An interval constraints can be used to find clusters which, for instance, contain only adjacent time points (i.e., which are continuous intervals), while a non-interval constraints can be used to find clusters which are not intervals. In the first case, we are able to capture associations which characterize any single stage of the sampling period, while in the second case, we might point out interactions which are somehow time-independent. Other interesting constraints might be defined like extended cannot-link or must-link constraints. Due to space limitation, this is out of the scope of this paper.

3 A Local-to-Global (L2G) Approach

In [9], a generic co-clustering framework is introduced and we have to recall it before its extension towards constraint-based co-clustering. The main idea is to compute bi-partitions from bi-sets which capture locally strong associations between sets of objects and sets of properties. Formally, a bi-set is an element $b_j = (T_j, G_j)$ ($T_j \subseteq \mathcal{T}$, $G_j \subseteq \mathcal{G}$) and we assume that a collection of a priori interesting bi-sets denoted $\mathcal{B}$ has been extracted from $\mathbf{r}$ beforehand. Let us now describe b_j by the Boolean vector $\langle \mathbf{t}_j \rangle, \langle \mathbf{g}_j \rangle = \langle t_{j1}, \dots, t_{jm} \rangle, \langle g_{j1}, \dots, g_{jn} \rangle$ where $t_{jk} = 1$ if $t_k \in T_j$ (0 otherwise) and $g_{jk} = 1$ if $g_k \in G_j$ (0 otherwise). We are looking for K clusters of bi-sets $\{P_1^{\mathcal{B}}, \dots, P_K^{\mathcal{B}}\}$ ($P_i^{\mathcal{B}} \subseteq \mathcal{B}$). Let us define the centroid of a cluster of bi-sets $P_i^{\mathcal{B}}$ as $\mu_i = \langle \tau_i \rangle, \langle \gamma_i \rangle = \langle \tau_{i1}, \dots, \tau_{im} \rangle, \langle \gamma_{i1}, \dots, \gamma_{in} \rangle$ where τ and γ are the usual centroid components:

$$\tau_{ik} = \frac{1}{|P_i^{\mathcal{B}}|} \sum_{b_j \in P_i^{\mathcal{B}}} t_{jk}, \quad \gamma_{ik} = \frac{1}{|P_i^{\mathcal{B}}|} \sum_{b_j \in P_i^{\mathcal{B}}} g_{jk}$$

We now define our distance between a bi-set and a centroid:

$$d(b_j, \mu_i) = \frac{1}{2} \left(\frac{|\mathbf{t}_j \cup \boldsymbol{\tau}_i| - |\mathbf{t}_j \cap \boldsymbol{\tau}_i|}{|\mathbf{t}_j \cup \boldsymbol{\tau}_i|} + \frac{|\mathbf{g}_j \cup \boldsymbol{\gamma}_i| - |\mathbf{g}_j \cap \boldsymbol{\gamma}_i|}{|\mathbf{g}_j \cup \boldsymbol{\gamma}_i|} \right)$$

It is the mean of the weighted symmetrical differences of the set components. We assume $|\mathbf{t}_j \cap \boldsymbol{\tau}_i| = \sum_{k=1}^m a_k \frac{t_{jk} + \tau_{ik}}{2}$ and $|\mathbf{t}_j \cup \boldsymbol{\tau}_i| = \sum_{k=1}^m \frac{t_{jk} + \tau_{ik}}{2}$ where $a_k = 1$ if $t_{jk} \cdot \tau_{ik} \neq 0$, 0 otherwise. Intuitively, the intersection is equal to the mean between the number of common objects and the sum of their centroid weights. The union is the mean between the number of objects and the sum of their centroid weights. These measures are defined similarly on properties.

Objects t_j (resp. properties g_j) are assigned to one of the K clusters (denoted i) for which τ_{ij} (resp. γ_{ij}) is maximum. We can enable that a number

Table 2. CDK-MEANS pseudo-code

CDK-MEANS ($\mathbf{r}$ is a Boolean context, $\mathcal{B}$ is a collection of bi-sets in $\mathbf{r}$, K is the number of clusters, MI is the maximal iteration number.)

1. Let $\mu_1 \dots \mu_K$ be the initial cluster centroids. $it := 0$.
 2. Repeat
 - (a) For each bi-set $b_i \in \mathcal{B}$, assign it to cluster $P_k^{\mathcal{B}}$ s.t. $d(b_i, \mu_k)$ is minimal.
 - (b) For each cluster $P_i^{\mathcal{B}}$, compute τ_i and γ_i .
 - (c) $it := it + 1$.
 3. Until centroids are unchanged or $it = MI$.
 4. For each $t_j \in \mathcal{T}$ (resp. $g_j \in \mathcal{G}$), assign it to the first cluster $P_i^{\mathcal{T}}$ (resp. $P_i^{\mathcal{G}}$) s.t. τ_{ij} (resp. γ_{ij}) is max.
 5. Return $\{P_1^{\mathcal{T}} \dots P_K^{\mathcal{T}}\}$ and $\{P_1^{\mathcal{G}} \dots P_K^{\mathcal{G}}\}$
-

of objects and/or properties belong to more than one cluster by controlling the size of the overlapping part of each cluster. Thanks to our definition of cluster membership determined by the values of τ_i and γ_i , we just need to adapt the cluster assignment step given some user-defined thresholds. A simplified algorithm CDK-MEANS from [9] is recalled in Table 2 (for the sake of brevity, we do not consider cluster overlapping). It computes a bi-partition of $\mathbf{r}$ given a collection of bi-sets $\mathcal{B}$ extracted from $\mathbf{r}$ beforehand (e.g., formal concepts). CDK-MEANS can provide the example bi-partition given in Section 1.

Let us now propose a significant extension of the L2G framework when an Interval or Non-interval constraint has been specified. The key idea is that, to compute a bi-partition which satisfies a (global) constraint, we can process a collection of local patterns which do not violate a local counterpart of this constraint. Using such a local level constraint (possibly associated with a propagation strategy), might enable to get efficiently a bi-partition which satisfies the global level one. Notice that given the state-of-the-art in constraint-based mining of bi-sets, quite efficient algorithms can now extract constrained bi-sets. For instance, in our applications, we use D-MINER [11] for computing complete collections of formal concepts which also satisfy various user-defined constraints. It is possible to enforce the same interval and non-interval constraints in the used bi-set collection. However, in the case of Interval constraint, it might be too stringent in practice, while for Non-interval, it will not be selective enough. For this reason, we propose to relax the Interval constraint and strengthen the Non-interval one on bi-sets by introducing two new local constraints.

Definition 2 (max-gap and min-gap constraints). *Given an order on $\mathcal{D}$, a max-gap constraint on this dimension, denoted $\mathcal{C}_{maxgap}(\mathcal{D}, l, b)$, is satisfied iff, for each pair of consecutive elements $x_i, x_j \in b$, $x_i \prec x_j$, $|\{x_h \notin b | x_i \prec x_h \prec x_j\}| \leq l$. A min-gap constraint, denoted $\mathcal{C}_{mingap}(\mathcal{D}, l, b)$, is satisfied iff, for each pair of consecutive elements $x_i, x_j \in b$, $x_i \prec x_j$, $|\{x_h \notin b | x_i \prec x_h \prec x_j\}| \geq l$.*

It is straightforward to prove the following property:

Property 1. *The min-gap constraint is anti-monotonic and can be used to efficient pruning.*

The max-gap constraint does not have any monotonicity property w.r.t. to set inclusion. In fact, for a dimension $D = \{x_1, x_2, \dots, x_n\}$ a max-gap constraint $\mathcal{C}_{maxgap}(D, 1)$, is not satisfied by the set $X_1 = \{x_2, x_3, x_7\}$, but is satisfied by its superset $X_2 = \{x_2, x_3, x_5, x_7\}$, and by its subset $X_0 = \{x_2, x_3\}$. Then it is neither monotonic nor anti-monotonic¹. The first one (max-gap) is used for the Interval constraint processing. The second one (min-gap) supports the Non-interval constraint processing. Clearly, final constraint satisfaction is not ensured, but the computational behavior is satisfactory (see the experimental section).

4 Experimental Validation

Evaluation Method. A general criterion to evaluate clustering results consists in comparing the computed partition with a “correct” one. It means that data instances are already associated to some correct labels and that one quantifies the agreement between computed labels and correct ones. A popular measure is the Rand index which measures the agreement between two partitions of m elements. If $\mathbf{C} = \{C_1 \dots C_s\}$ is our clustering structure and $\mathbf{P} = \{P_1 \dots P_t\}$ is a predefined partition, each pair of data points is either assigned to the same cluster in both partitions or to different ones. Let a be the number of pairs belonging to the same cluster of $\mathbf{C}$ and to the same cluster of $\mathbf{P}$. Let b be the number of pairs whose points belong to different clusters of $\mathbf{C}$ and to different clusters of $\mathbf{P}$. The agreement between $\mathbf{C}$ and $\mathbf{P}$ can be estimated using

$$Rand(\mathbf{C}, \mathbf{P}) = \frac{a + b}{m \cdot (m - 1)/2}$$

which takes values between 0 and 1 and is maximized when $s = t$.

We also want to evaluate co-clustering quality by means of an internal criterion. An interesting measure for this purpose is the symmetrical Goodman and Kruskal’s τ coefficient [12] which evaluates the proportional reduction in error given by the knowledge of C^o on the prediction of C^p and vice versa. It is evaluated in a contingency table $\mathbf{p}$. Let p_{ij} be the frequency of relations between an object of a cluster C_i^o and a property of a cluster C_j^p , and $p_{i.} = \sum_j p_{ij}$ and $p_{.j} = \sum_i p_{ij}$. The Goodman-Kruskal’s τ coefficient, is defined as follows:

$$\tau = \frac{\frac{1}{2} \sum_i \sum_j (p_{ij} - p_{i.} p_{.j})^2 \frac{p_{i.} + p_{.j}}{p_{i.} p_{.j}}}{1 - \frac{1}{2} \sum_i p_{i.}^2 - \frac{1}{2} \sum_j p_{.j}^2}$$

Time Interval Cluster Discovery. We have studied the impact of the Interval constraint in two microarray data sets, *malaria* and *drosophila*. The first one

¹ The search space can be pruned by considering specific ordering at candidate generation phase.

[13] concerns the transcriptome of the intraerythrocytic developmental cycle of *Plasmodium Falciparum*, i.e., a causative agent of human malaria. The data provide the expression profile of 3 719 genes in 46 biological samples. Each sample corresponds to a time point of the developmental cycle: it begins with merozoite invasion of the red blood cells, and it is divided into three main phases, the ring, trophozoite and schizont stages. The second data set is described in [14]. It concerns the gene expression of the *Drosophila melanogaster* during its life cycle. The expression levels of 3 944 genes are evaluated for 57 sequential time periods divided into embryonic, larval and pupal stages. The numerical gene expression data given in [13] has been discretized by using one of the encoding methods described in [15]: for each gene g , we assigned the Boolean value 1 to those samples whose expression level was greater than $X\%$ of its max expression level. X was set to 25% for **malaria** and 35% for **drosophila**. The two matrices have been mined for formal concepts by using D-MINER [11].

We applied COCLUSTER algorithm [2], and the unconstrained version of CDK-MEANS with $K = 3$ to identify the three developmental stages. Since the initialization of both algorithms is randomized, we average all the measures on 100 executions. We have measured the Rand index w.r.t. to the real partitioning (which has been inferred from the literature), and the Goodman-Kruskal's coefficient to evaluate the bi-partition quality.

There is a significant difference between the two data sets. In **malaria**, if COCLUSTER achieves a good Goodman-Kruskal's coefficient, the bi-clusters obtained by CDK-MEANS are more consistent with the biological knowledge (i.e., the partition has a higher Rand index). On another hand, the number of comparisons is rather high. What we expect here, is that a constrained approach can obtain the same clustering results by using less computing resources. Instead, for **Drosophila**, both algorithms fail in finding the correct partitioning w.r.t. the available biological knowledge. The number of jumps is in both cases high, while the Rand index is relatively low. In this case we expect to obtain better results with our constrained clustering approach.

We have defined the Interval constraint on the biological condition dimension. Different levels of the max-gap constraint have been applied and we have studied the impact on the final partition by measuring the Rand index and the Goodman-Kruskal's coefficient.

For **malaria** (graphics are omitted for sake of brevity), for low values of max-gap, we obtain a better agreement w.r.t. to the three developmental stages, while the Goodman-Kruskal's coefficient is not significantly dissimilar to the one obtained without constraints. On another hand, setting a max-gap constraint considerably reduce the size of the collection, then CDK-MEANS perform faster. As a secondary observation, notice that our definition of max-gap constraint works for open time intervals. By setting an open time interval constraint, we are always able to obtain a circular sequence of intervals (capturing typical developmental life cycles).

For **drosophila**, the improvements are more obvious. Unconstrained clustering results have shown that good partitions (with a high Goodman-Kruskal's

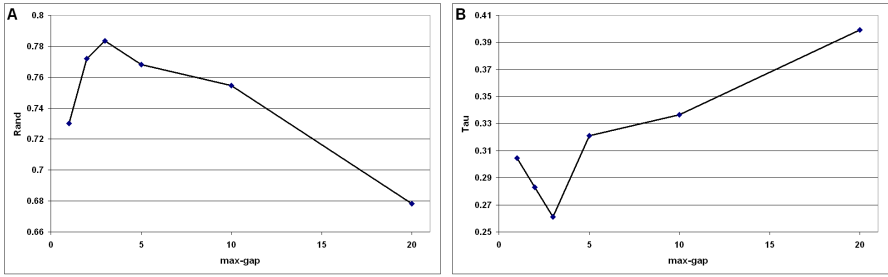


Fig. 1. Results on the *drosophila* dataset

coefficient) contain a lot of jumps. With a max-gap constraint of 2 or 3, we can sensibly increase the quality of the partition (Fig. 1a) w.r.t. the available biological knowledge. The fact that for these max-gap values, the Goodman-Kruskal's coefficient is minimum (Fig. 1b), indicates that the partition which better satisfies the constraints is not necessarily the “best” one. Moreover, the average number of comparisons is reduced by 60 (max-gap=2) and 30 (max-gap=3).

Using Non-interval Constraint. We have shown how interval constraints can support the discovery of time interval clusters. Within some data (e.g., *malaria*), an unconstrained approach already gives perfect intervals, and then the question is: is it possible to discover different gene associations which hold between time points belonging to different intervals? To answer this question, we applied the Non-interval constraint to the gene expression data concerning adult time samples of the *drosophila melanogaster* life cycle. Time samples from t_1 to t_{10} concern the first days of male adult individual life cycle. Time samples from t_{11} to t_{20} concern female individuals.

When we apply CDK-MEANS (with $k = 2$) without specifying any constraint on this data set, the two intervals $t_1, \dots, t_{10}$ and $t_{11}, \dots, t_{20}$ are well identified in the 100 executions of the algorithm. Then, we obtain almost exactly a bi-cluster

Table 3. Clustering results on adult *drosophila* individuals

bi-part.	inst.	τ		<i>Rand</i>	
		mean	std.dev	mean	std.dev
co:MF	56	0.5605	0.0381	0.82	0.06
co:mixed	44	0.1156	0.0166	0.51	0.02
co:overall	100	0.3648	0.2240	0.69	0.16
cdk:unconst	100	0.4819	0.0594	0.88	0.04
cdk:int	100	0.4609	0.0347	1.00	0.00
cdk:nonint	100	0.1262	0.0761	0.53	0.04

of males and a bi-cluster of females. Moreover, the Goodman-Kruskal's coefficient and the loss in mutual information appears rather stable (see `cdk:unconst` result on Tab. 3). We computed these coefficients on the 100 bi-partitions returned by COCLUSTER and we noticed a significant unstability (see Tab. 3). It seems that there are two optimum points for which the two measures are distant. For 56 runs, we got a high τ coefficient (mean 0.5605), for the other 44 ones the τ coefficient was sensibly smaller (mean 0.1156). If we consider each group of results separately, the standard deviation is significantly smaller. It means that these two results are two local optima for the COCLUSTER heuristics. From a semantical point of view, the first group of solution reflects the male and female repartition of the individuals, while in the second group each cluster contains both male and female individuals. The average Rand value is 0.69 and the standard deviation is 23% of the mean. Then, we tried to specify a min-gap constraint on the collection of formal concepts. Even for small values of the min-gap constraint, the average Rand value is high, while the standard deviation is lower (12% of the mean for min-gap=2, 4% for min-gap=3) w.r.t. COCLUSTER results. The `cdk:nonint` row in Tab. 3 summarizes the more stable (w.r.t. the τ coefficient) results obtained with min-gap=10. We also tested whereas an interval constraint could influence the stability of the bi-partition. Setting max-gap=5 enables to get more stable bi-partitions where the Rand index is always equal to one (see `cdk:int` results in Tab. 3). These results show that, by specifying an Interval or a Non-interval constraint, the user gets some control on the shape of the bi-partition. An algorithm like COCLUSTER has sometimes found bi-clusters where the sex of the individual is the major discriminative parameter. At some moment, it has captured something else. Our thesis is that a biologist might be able to have a kind of supervision on such a process. Moreover, using constraints also speeds up the bi-partition construction because we have to process a reduced collection of bi-sets.

5 Conclusion

Co-clustering is an interesting conceptual clustering approach. Improving bi-cluster relevancy remains a difficult task in real-life exploratory data analysis processes. First, it is hard to capture subjective interestingness aspects, e.g., the analyst's expectation given her/his domain knowledge. Next, when these expectations can be declaratively specified, using them during the computational process is challenging. We have shown that it was possible to use a simple but powerful generic bi-clustering framework based on local patterns. New types of constraints on bi-clusters have been considered when at least one of the dimensions is ordered. Applications on temporal gene expression data analysis have been sketched. Many other applications rely on ordered data analysis and might benefit from such constrained co-clustering approaches. A short-term perspective is to formalize the properties of the global constraints (i.e., constraints on bi-partitions) which can be, more or less automatically, transformed into local level constraints.

Acknowledgements. This research is partially funded by ACI MD 46 BINGO and by EU contract IQ FP6-516169 (FET arm of the IST programme).

References

1. Robardet, C., Feschet, F.: Efficient local search in conceptual clustering. In: Proceedings DS'01. Volume 2226 of LNCS., Washington, USA, Springer-Verlag (2001) 323–335
2. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: Proceedings ACM SIGKDD 2003, Washington, USA, ACM Press (2003) 89–98
3. Ritschard, G., Zighed, D.A.: Simultaneous row and column partitioning: Evaluation of a heuristic. In: Proceedings of the 14th International Symposium ISMIS 2003. Volume 2871 of LNCS., Maebashi City, Japan, Springer (2003) 468–472
4. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **1**(1) (2004) 24–45
5. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: Proceedings ICML 2001, Williamstown, USA, Morgan Kaufmann (2001) 577–584
6. Klein, D., Kamvar, S.D., Manning, C.D.: From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: Proceedings ICML 2002, Sydney, Australia, Morgan Kaufmann (2002) 307–314
7. Davidson, I., Ravi, S.S.: Clustering with constraints: Feasibility issues and the k-means algorithm. In: Proceedings SIAM SDM 2005, Newport Beach, USA (2005)
8. Davidson, I., Ravi, S.S.: Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In: Proceedings PKDD 2005. Volume 3721 of LNCS., Porto, Portugal, Springer (2005) 59–70
9. Pensa, R.G., Robardet, C., Boulicaut, J.F.: A bi-clustering framework for categorical data. In: Proceedings PKDD 2005. Volume 3721 of LNAI., Porto, Portugal, Springer-Verlag (2005) 643–650
10. Sese, J., Kurokawa, Y., Monden, M., Kato, K., Morishita, S.: Constrained clusters of gene expression profiles with pathological features. *Bioinformatics* **20**(17) (2004) 3137–3145
11. Besson, J., Robardet, C., Boulicaut, J.F., Rome, S.: Constraint-based concept mining and its application to microarray data analysis. *Intelligent Data Analysis* **9**(1) (2005) 59–82
12. Goodman, L.A., Kruskal, W.H.: Measures of association for cross classification. *Journal of the American Statistical Association* **49** (1954) 732–764
13. Bozdech, Z., Llinás, M., Pulliam, B.L., Wong, E., Zhu, J., DeRisi, J.: The transcriptome of the intraerythrocytic developmental cycle of *plasmodium falciparum*. *PLoS Biology* **1**(1) (2003) 1–16
14. Arbeitman, M., Furlong, E., Imam, F., Johnson, E., Null, B., Baker, B., Krasnow, M., Scott, M., Davis, R., White, K.: Gene expression during the life cycle of *drosophila melanogaster*. *Science* **297** (2002) 2270–2275
15. Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J.F., Gandrillon, O.: Strong association rule mining for large gene expression data analysis: a case study on human SAGE data. *Genome Biology* **12** (2002)

A Comparative Analysis of Clustering Methodology and Application for Market Segmentation: K-Means, SOM and a Two-Level SOM

Sang-Chul Lee¹, Ja-Chul Gu², and Yung-Ho Suh²

¹ Department of Management Information Systems, Korea Christian University,
San 204, Hwagok 6-Dong, Kangseo-Ku, Seoul 157- 722, South Korea
leecho@kcw.ac.kr

² School of Business Administration, Kyung Hee University,
1 Hoegi-Dong, Dongdaemoon-Gu, Seoul 130-701, South Korea
{moonlight, suhy}@khu.ac.kr

Abstract. The purpose of our research is to identify the critical variables, to evaluate the performance of variable selection, to evaluate the performance of a two-level SOM and to implement this methodology into Asian online game market segmentation. Conclusively, our results suggest that weight-based variable selection is more useful for market segmentation than full-based and SEM-based variable selection. Additionally, a two-level SOM is more accurate in classification than K-means and SOM. The critical segmentation variables and the characteristics of target customers were different among countries. Therefore, online game companies should develop diverse marketing strategies based on characteristics of their target customers using research framework we propose.

1 Introduction

To survive successfully in today's volatile and competitive game markets, online game companies need to determine who the target customers are and what motivates them. This process is called market segmentation, by which companies are able to understand their loyal customers and concentrate their limited resources into them [17]. Given the importance of market segmentation, the primary issues are twofold; variable selection (which variables are selected) and clustering methodologies (which methodology is best for market segmentation).

First, the objective of variable selection is improving the classification performance of the separators. Therefore, variable selection has become the focus of much research in areas of application for market segmentation [12]. However, variables of previous online game research could not be accepted since they have been conducted mainly from the technological perspective. The main concern of technological research on online game is to design and develop a more attractive and effective online game environment [1, 20, 26]. However, no matter how sophisticated the technologies applied, users would not revisit the game site if it failed to reflect their needs. Therefore, our research identifies the primary factors for online game from a business perspective.

Additionally, another problem is which variables are useful to build a good separator. To select the variables, application research use Structural Equation Model (SEM) in order to find which amongst them bore maximum predictive power as to predict the dependent variable [23]. Data mining research use the weight-based algorithm to reflect customers' preference by allowing different weights [16]. To evaluate the performance of variable selection, our research use SEM and the weight-based algorithm, as viewed in the context of the behavioral research using the observable variable. In experiments, we use three data sets; full-based variable selection (all variables are included), SEM-based variable selection (insignificant variables are eliminated) and weight-based variable selection (variables are weighted by coefficient of SEM).

Second, the traditional methodology for market segmentation was based mainly on statistical clustering techniques; hierarchical and partitive approaches. However, hierarchical method can not provide a unique clustering because a partitioning to cut the dendrogram at certain level is not precise. This method ignores the fact that the within-cluster distance may be different for different clusters [7], [24]. Partitive method predefines the number of clusters, before performing it. It can be part of the error function and can not identify the precise number of clusters [8], [19], [24]. Additionally, these algorithms are known to be sensitive to noise and outliers [5], [6], [24].

During the last years, neural network models are proposed to settle these problems. One of the methodologies is a two-level Self-Organizing Map (SOM): SOM training and clustering [24]. Instead of clustering the data directly, a large set of prototypes is formed using the SOM. The prototypes can be interpreted as proto-cluster, which are combined in the next phase to form the actual clusters. The benefit of this method is the reduction of the computational cost and the noise [24].

Although a two-level SOM have been proposed, little comparison with traditional methodologies and little industrial application are available. Therefore, our research evaluates the performance of three clustering methodologies, K-means, SOM and a two-level SOM and implements this method into marketing research field.

The purpose of our research is to identify the critical variables, to evaluate the performance of variable selection, to evaluate the performance of a two-level SOM and to implement this methodology into Asian online game market segmentation. To implement our methodology, Korean (KR), Japanese (JP) and Chinese (CH) online game data were analyzed because they were located in the center of those trends. Therefore, our research will be helpful for other countries to understand the change of Asian and global game markets.

2 Determinant Variables

Through the review of the relevant literature, we identify the primary factors for online game from a business perspective as follows: the convenience of operator, the suitability of feedback, the reality of design, the precision of information and the involvement of virtual community. Our research hypothesizes that these determinants have a positive effect on flow.

The convenience of the operator was defined as the manipulatability of operators to play games [21]. Operator is an important determinant of influencing interaction between users and games [2], [11], [25]. Feedback is the reaction from online games [4], [9]. For example, when players kill a monster within NCsoft's Lineage, they receive feedback upgrading their level. The reality of design is defined as the design of interface making gamers feel online games as part of the real world [1], [20], [27]. Information is the contents from online game to achieve the stated goals. Gamers who received more precise information about how to play the games tended to achieve online game goals and experience flow easier [9], [18]. Virtual community is defined as computer-mediated spaces with potential for integration of member-generated content and communication [13]. Online game users should solve problems together interacting with other users in virtual communities [9].

3 A Two-Level SOM

Vesanto and Alhoniemi [24] proposed a two-level SOM: SOM training and clustering. A two-level SOM was combined SOM, K-means and DB Index. In the first level (SOM training), the data were clustered directly in original SOM to form a large set of prototypes. In the second level (SOM clustering), the prototypes of SOM are clustered using k-means. Finally, to select the best one among different partitioning, a two-level SOM used DB index. Among several validity indices of clustering methodology, a two-level SOM used DB index because it was suitable for evaluation of k-means partitioning [21]. Conclusively, the proper clustering is achieved by minimizing the DB index [10]. The procedure of a two-level SOM is shown in Fig.1.

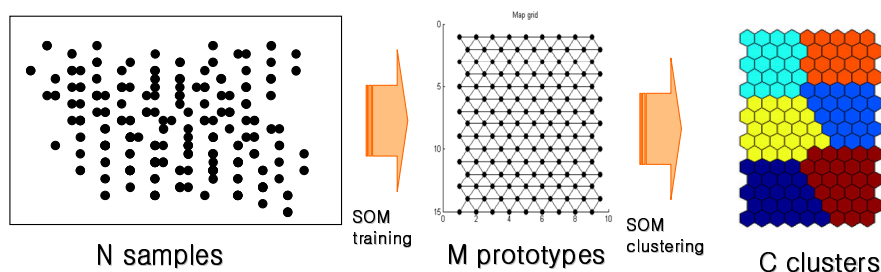


Fig. 1. A two-level SOM

4 Research Framework

We propose a research framework for market segmentation, as viewed in the context of SEM and a two-level SOM. To segment the online game market and develop marketing strategies, our research approach is categorized into two phases. First, we perform the Confirmatory Factor Analysis (CFA) and SEM for Korean, Japanese, and Chinese samples to identify the critical segmentation variables. Although neural

networks (SOM) have established themselves as a standard analytical tool in many business areas, they are used to carry out as part of a *tandem approach*, which involves the factor analysis and multiple regression of the observable variables prior to clustering [23]. In this section, we also evaluate the performance of variable selection.

Second, we perform a two-level SOM to segment online game market using weight-based variables, which were given the different weights of the path coefficient. In this section, a two-level SOM is compared with traditional methodologies.

Final, after segmentation of the markets, we use ANOVA and cross tabulation analysis to recognize the characteristics of sub-divided clusters. Conclusively, we target a segment market with the highest customer loyalty and compare the target customers in Korea, Japan, and China.

5 Results

5.1 Data and Measurement

To test the model, a convenience sample of 1704 (KR), 602 (JP), and 592 (CN) online game users were available for analysis, after elimination of missing data. Our research developed multi-item measures for each construct. Twenty-one items for five determinants are selected. We asked respondents to indicate on a five (KR/JP) and seven (CH) point Likert scale to what extent the determinants influence on flow in online game. We used CFA to evaluate convergent validity for five constructs. The results indicated that 15 items for five determinants remained within Korean and Japanese model. However, 11 items for four determinants remained for Chinese model, eliminating the suitability of feedback because it was not significant. All the fit statistics of the measurement model were acceptable.

5.2 Identification of Critical Variables

To find the critical factors for segmentation, we used AMOS 5.0 in SEM. The structural model of Korea, Japan and China was well converged. The results indicated that the chi-square of the model was 201.01(KR)/ 215.55(JP)/ 146.11(CN) with d.f. of 104(KR)/ 104(JP)/ 55(CN), the ratio of chi-square to d.f. was 2.702(KR)/ 2.073(JP)/ 2.656(CN), GFI was 0.981(KR)/ 0.959(JP)/ 0.965 (CN), AGFI was 0.972(KR)/ 0.940(JP)/ 0.942(CN) and RMSR was 0.019(KR)/ 0.040(JP)/ 0.074(CN); all the fit statistics were acceptable.

The results of SEM indicated that the significant variables for market segmentation were different among each nation. For Korean market, four of the five paths were statistically significant and the path from the convenience of operator to flow was insignificant, as shown in Table 1. For Japanese market, four of the five paths were statistically significant and the path from the suitability of feedback to flow was insignificant. For Chinese market, three of the four paths were statistically significant and the path from the involvement of virtual community to flow was insignificant. Especially, the precision of information influenced negatively.

Table 1. The results of Stuctural Equeation Model

	Path	Korea	Japan	China
O	-->	0.030	0.273**	0.072*
FB	-->	0.116**	0.058	-
IF	-->	0.079*	0.136*	-0.312**
D	-->	0.283**	0.160**	0.236**
C	-->	0.417**	0.385**	0.001

* $p < 0.05$, ** $p < 0.01$

O: The convenience of operator,

FB: The suitability of feedback

IF: The precision of information,

D: Reality of Design

C: The involvement of virtual community,

F: Flow

In comparison with Korea and Japan, the convenience of operator was not significant but the suitability of feedback was significant in Korean model, in contrast to Japanese model. It was interpreted that Korean online gamers preferred to be reacted appropriately and faster when they completed their missions, because they wanted to achieve a high status in virtual community. Conversely, the Japanese gamers preferred to grow their characters at their convenience. This hypothesis was proven in the case of “Vandai’s damakuchi”, which is the game growing animal characters for a long time [15]. In China, the precision of information was negative influenced. Recently, more than 70% of Chinese online games were foreign. However, most information provided by these online games was mistranslated and incorrect so that gamer had to obtain precise information from other channels, such as online game magazine and community sites [14]. Therefore, Chinese gamers did not like incorrect information provided when they played online game and showed negative attitude to the provision of information.

5.3 Comparison of Variable Selection

To evaluate the performance of variable selection in the observable variables, we used a two level SOM in conjunction with DB index. In the experiments, three data sets were used. The data set of full-based variable selection consisted of all variables. For Korean example, five variables were included. The data set of SEM-based variable selection consisted of significant variables. For Korean example, four variables were included except the convenience of operator. The data set of weight-base variable selection consisted of variables given the weight of the path coefficients.

Table 2. The results of comaprision of variable selection

	full-based		SEM-based(4)		weight-based(4)	
	mean	DB Index	mean	DB Index	mean	DB Index
Korea	0.801	0.722	0.793	0.702	0.739	0.650
Japan	0.784	0.712	0.766	0.670	0.756	0.658
China	0.785	0.717	0.754	0.669	0.737	0.663

The mean and DB index value of three data sets in Korea, Japan and China were shown in Table.2. Especially, the variation of DB index value in Korean data sets was shown in Fig.2. The results of comparison analysis indicated that the mean and DB index value of the weight-based variable selection were lower than others in Korea, Japan and China data sets. Therefore, the variable given the weight value has more performance than full-based and SEM-based variable selection.

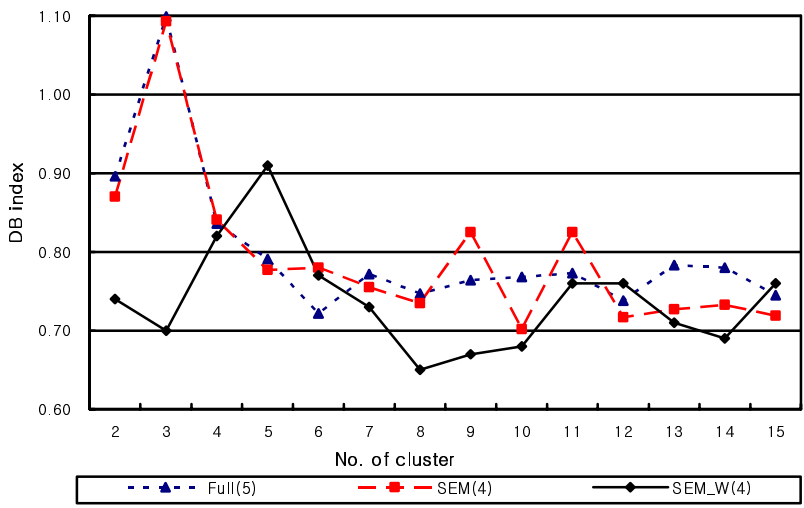


Fig. 2. The DB Index of full-based variable selection (5), SEM-based variable selection (4), and Weight-based variable selection (4) in Korea data set

5.4 Market Segmentation

To segment the Korean online game market, our research was conducted using a two-level SOM. In the experiments, the first level was SOM training. 1704 (KR)/ 602(JP)/ 592(CN) data samples and 4(KR)/ 4(JP)/ 3(CN) significant variables were used. A SOM was trained using the sequential training algorithm for data samples. A neighborhood width decreased linearly 5 to 1 using the Gaussian function. A map was used by 20*10 (KR)/ 13*9 (JP)/ 13*9 (CN)/ matrix and 209(KR)/ 120(JP)/ 117(CN) prototypes were developed.

The second level was SOM clustering. The partitive clustering of 210(KR)/ 117(JP)/ 117(CN) SOM's prototypes was carried out using batch K-means algorithm. The K-means ran multiple times for each k. The DB index was used to select the best clustering. The analysis of the DB index resulted in the development of eight (KR)/ six (JP)/ six (CN) market segments.

5.5 Comparison of Clustering Methodologies

To evaluate the performance of three clustering methodologies, K-means, SOM and a two-level SOM, we used a two level SOM in conjunction with DB index and

ANOVA (Analysis of Variance). The results of comparison analysis indicated that the DB index value of a two-level SOM was lower than others in Korea, Japan and China data sets in Fig.3.

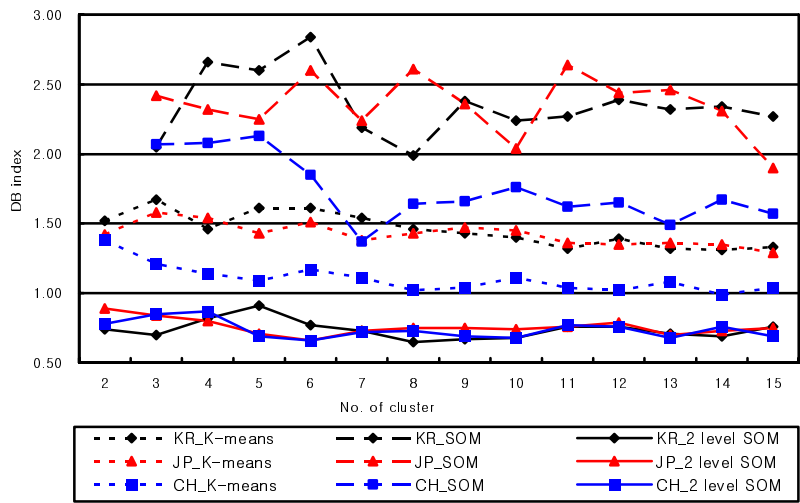


Fig. 3. DB Index of K-means, SOM and a two-level SOM in Koran, Japanese and China data set

Additionally, we used ANOVA to test for significant differences among K-means, SOM and a two-level SOM. According to results of ANOVA, the difference of all methodologies was significant in Table.3. Therefore, a two-level SOM is more accurate in classification than K-means and SOM.

Table 3. ANOVA of clusters

	K-means	SOM	a two-level SOM	F	Sig.
	mean	mean	mean		
Korea	1.455	2.349	0.739	362.23	0.00
Japan	1.423	2.353	0.756	472.80	0.00
China	1.103	1.738	0.737	153.02	0.00

5.6 Determination of Target Market

After segmenting the markets, online game companies should evaluate the attractiveness of each segment and select the target segment. To identify target market, we used ANOVA to recognize the customer loyalty of each cluster. The results indicated that the target market was cluster 7 (KR)/ cluster 3 (JP)/ cluster 5

(CN) among their clusters in Table 4. The other analysis of the intention of revisit and WOM (Word of Mouth) indicated the same results.

To identify the structure of the clusters, we categorized the effectiveness of the variables into 3 levels; high, middle and low. Additionally, we conducted on the analysis of the demographic and behavioral variables using cross tabulation analysis: gender, age, job, school, income level, i_year (how long did gamers use the Internet), i_day (how many hours did gamer use the Internet per day), and g_day (how many hours did gamer play online games per day). The characteristics and structure of clusters are summarized in Table 4.

Table 4. Profiles of target market and negative market

Cluster	Target Market			Negative Market		
	Korea	Japan	China	Korea	Japan	China
	C 7 (n=270)	C 3 (n=103)	C 5 (n=157)	C 1 (n=223)	C 6 (n=88)	C 3 (n=100)
O	-	3.75 (H)	3.54 (H)	-	2.58 (L)	2.96 (M)
FB	2.93 (M)	-	-	2.43 (L)	-	-
IF	3.48 (H)	2.91 (M)	2.09 (L)	2.71 (L)	2.27 (L)	2.94 (M)
D	3.77 (H)	3.62 (H)	3.82 (H)	2.62 (L)	2.82 (M)	2.69(L)
C	4.17 (H)	3.62 (H)	-	3.01 (M)	2.52 (L)	-
Gender	female	male	male	male	male	Male
Age	20-30	36-	17-18	26-30	36-	19-21
Job	Student	Employee	Student	Student	Employee	Student
School	High School Graduate	High School Graduate	High School	University Graduate	High School Graduate	University
Income (\$)	-1,000	-1,000	- 50	1,001- 2,000	-1,000	-50
i_year	2-4	3	1-2	2-4	2-5	5-6
i_day	5	3-5	0-1	2-5	2-3	4-5
G_day	1-2	1	0-1	1	1	1-2
Revisit	4.04	4.10	3.76	3.03	3.00	2.79
WOM	4.05	3.62	3.69	2.90	2.38	2.95
Loyalty**	4.04	3.86	3.72	2.96	2.69	2.87

* L=Low, M=Middle, H=High

The results indicated that the critical segmentation variables and the characteristics of loyal customers were different among each nation. The main variable was the involvement of virtual community (KR)/ the convenience of operator (JP/CN). As to the demographic information, gender of the Korean primary target market was female while that of the Japanese and Chinese markets was male. An age of the Korean markets was 20-30 years while the Japanese market was over 36 years and Chinese market was under 17-18. Therefore, companies should develop strategies depending on the effectiveness of the variables and the demographic and behavioral characteristics of target market.

6 Conclusion and Limitation

With the rapid growth of Asian online game markets, many online game companies hoped that the first mover would be successful and recklessly entered into online game markets without understanding the core needs of those audiences. However, the lack of consideration has forced many online game companies to fail to survive in game market [15].

To survive in today's competitive markets, online game companies need to understand their loyal customers and concentrate their limited resources. This raises an interesting question about which clustering methods should be adopted and which variables are useful in the development of marketing strategies [3]. Our research was performed to evaluate the performance of variable selection and to evaluate the performance of a two-level SOM. Additionally, our research proposed a research framework for implementing this methodology into marketing research field and applied to Asian online game markets.

The results of comparison indicated that the weight-base variable selection and a two-level SOM have more performance than others as shown in section 5.3 and 5.5. Additionally, the results of market segmentation indicated that the critical segmentation variables and the characteristics of loyal customers were different among each nation as shown in section 5.6. Therefore, countries with different cultural and industrial background might have to be very careful about developing their own marketing strategies using a conceptual framework we propose.

References

1. Ackley, J.: Roundtable Reports: Better Sound Design. Gamasutra. (1998). http://www.Gamasutra.com/features/gdc_reports/cgdc_98/ackley.htm
2. Agarwal, R., Karahanna, E.: Time Files When You're Having Fun: Cognitive Absorption and Beliefs About Information Technology Usage. *MIS Quarterly*, Vol.24, No.4. (2000) 665-694
3. Balakrishnan, P. V., Cooper, M. C., Jacob, V. S., Lewis, P. A.: Comparative performance of the FSCL Neural Net and K-means Algorithm algorithm for market segmentation. *European Journal of Operational Research*, Vol.93 (1996) 346-357
4. Baron, J.: Glory and Shame: Powerful Psychology in Multiplayer Online Games. Gamasutra. (1999). http://www.gamasutra.com/features/19991110/Baron_01.htm
5. Bezdek, J. C.: Some New Indexes of Cluster Validity. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol.28, No.3. (1998) 301-315
6. Blatt, M., Wiseman, S., Domany, E.: Super-paramagnetic Clustering of Data. *Physical Review Letters*, Vol.76, No.8. (1996) 3251-3254
7. Boudaillier, E., Hebrail, G.: Interactive Interpretation of Hierarchical Clustering. *Intelligent Data Analysis*, Vol.2, No.3. (1998) 41-
8. Buhmann, J., Kühnel, H.: Complexity Optimized Data Clustering by Competitive Neural Networks. *Neural Computation*, Vol.5, No.3. (1993) 75-88
9. Choi, D. S., Park, S. J., Kim, J. W.: A Structured Analysis Model of Customer Loyalty in Online Games. *Journal of MIS Research*, Vol.11, No.3. (2001) 1-20
10. Davies, D. L., Bouldin, D. W.: A Cluster Separation Measure. *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol.1. (1979) 224-227

11. Davies, F. D., Bagozzi, R. P., Warshaw, P. R.: Extrinsic and Intrinsic Motivation to Use Computers in the Workplace. *Journal of Application Society Psychology*, Vol. 22, No. 14. (1992) 1111-1132
12. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, Vol.3. (2003) 1157-1182
13. Hagel, J., Armstrong, A.: *Net Gain: Expanding Markets through Virtual Communities*. Harvard Business School Press, Boston. (1997)
14. iResearch China Online Game Users Survey Report 2003, Shanghai, 2003.
15. KGDI: 2003 Korean White Game Paper. KGDI, Seoul. (2003)
16. Kim, C. Y., Lee, J. K., Cho, Y. H., Kim, D. H.: Viscors: A Visual-Content Recommender for the Mobile Web. *IEEE Intelligent Systems*, Vol.19. No.6. (2004) 1541-1672
17. Kotler, P.: *Marketing Management: Analysis, Planning, Implementation and Control*, 9th edn. A Simon and Schuster Co, New Jersey. (1997)
18. Lewinski, J. S.: *Developer's Guide to Computer Game Design*. Wordware Publishing Inc, Texas. (2000)
19. Maulik, U., Bandyopadhyay, S.: Performance Evaluation of Some Clustering Algorithms and Validity Indices. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, Vol. 24, No.12. (2002) 1650-1654
20. Sanchez-Crespo, D.: 99 from a Game Development Perspective. Gamasutra. (1999). http://gamasutra.com/features/19990802/siggraph_01.html
21. Shim, Y. S., Chung, J. W., Choi, I. C.: A Performance Comparison of Cluster Validity Indices based on K-means Algorithm. *Journal of MIS Research*, Vol.16, No.1. (2006) 127-144
22. Spector, W.: Remodeling RPGs for the New Millennium. Gamasutra. (1999). http://www.gamasutra.com/features/game_desing/19990115/remodeling_01.htm
23. Vellido, A., Lisboa, P.J.G., Meehan, K.: Segmentation of the on-line Shopping Market using Neural Networks. *Expert Systems with Applications*, Vol.17. (1999) 303-314
24. Vesanto, J., Alhoniemi, E.: Clustering of the Self-organizing Map. *IEEE Transactions on Neural Networks*, Vol.11, No.3. (2000) 586-600
25. Webster, J., Martocchio, J. J.: Micro Computer Playfulness: Development of a Measure with Workplace Implications. *MIS Quarterly*, Vol.16, No.2. (1992) 201- 226
26. Wells, J. D., Fuerst, W. L., Choobineh, J.: Managing Information Technology for One to One Customer Interaction. *Information and Management*, Vol.35, No.1. (1999) 53-62
27. Woodcock, W.: Game AI: the State of the Industry. Gamasutra. (1999). http://www.gamasutra.com/features/19990820/game_ai_01.html

Action Rules Discovery, a New Simplified Strategy

Zbigniew W. Raś^{1,2} and Agnieszka Dardzińska³

¹ UNC-Charlotte, Department of Computer Science, Charlotte, N.C. 28223, USA

² Polish Academy of Sciences, Institute of Computer Science,
Ordona 21, 01-237 Warsaw, Poland

`ras@uncc.edu`

³ Bialystok Technical Univ., Dept. of Mathematics, ul. Wiejska 45A, 15-351
Bialystok, Poland

`adardzin@uncc.edu`

Abstract. A new strategy for discovering action rules (or interventions) is presented in this paper. The current methods [14], [12], [8] require to discover classification rules before any action rule can be constructed from them. Several definitions of action rules [8], [13], [9], [3] have been proposed. They differ in the generality of their classification parts but they are always constructed from certain pairs of classification rules. Our new strategy defines the classification part of an action rule in a unique way. Also, action rules are constructed from single classification rules. We show how to compute their confidence and support. Action rules are used to reclassify objects. In this paper, we propose a method for measuring the level of reclassification freedom for objects in a decision system.

1 Introduction

There are two aspects of interestingness of rules that have been studied in data mining literature, objective and subjective measures [5], [1], [10], [11]. Objective measures are data-driven and domain-independent. Generally, they evaluate the rules based on their quality and similarity between them. Subjective measures, including unexpectedness, novelty and actionability, are user-driven and domain-dependent. A rule is actionable if user can do an action to his/her advantage based on this rule [5]. A new class of rules (called action rules) constructed from certain pairs of association rules, proposed in [8], refers to actionability in a more precise way. A formal definition of an action rule was independently proposed in [2]. Also, interventions introduced in [3] are conceptually very similar to action rules. These rules have been investigated further in [12], [13], [14], [9].

To give an example justifying the need of action rules, let us assume that a number of customers have closed their accounts at one of the banks. We construct, possibly the simplest, description of that group of people and next search for a new description, similar to the one we have, with a goal to identify a new group of customers from which no-one left that bank. If these descriptions have a

form of rules, then they can be seen as actionable rules. Now, by comparing these two descriptions, we may find the cause why these accounts have been closed and formulate an action which if undertaken by the bank, may prevent other customers from closing their accounts. Such actions are stimulated by action rules and they are seen as precise hints for actionability of rules. For example, an action rule may define a group of customers who are seriously thinking to close their bank accounts but if they are invited by the bank for a glass of wine and also get certain offers, most probably they will stay.

The current definition of action rules requires constructing them from certain pairs of classification rules. This assumption makes the process of action rules discovery not only unnecessarily expensive but also gives too much freedom in constructing their classification parts. This paper presents a new strategy for action rules construction and shows how they relate to action rules introduced in [9], [13].

2 Action Rules Discovery

An information system is used to store information about a group of related objects. Its definition, given here, is due to Pawlak [6]. By an information system we mean a pair $S = (U, A)$, where:

- U is a nonempty, finite set of objects (object identifiers),
- A is a nonempty, finite set of attributes (functions) i.e. $a : U \rightarrow V_a$ for $a \in A$, where V_a is called the domain of a .

We often write (a, v) instead of v , assuming that $v \in V_a$. Information systems can be seen as decision tables. In any decision table together with the set of attributes a partition of that set into conditions and decisions is given. Additionally, we assume that the set of conditions is partitioned into stable and flexible conditions [8]. Attribute $a \in A$ is called stable for the set U , if its values assigned to objects from U can not change in time. Otherwise, it is called flexible. "*Date of Birth*" is an example of a stable attribute. "*Interest rate*" on any customer account is an example of a flexible attribute. For simplicity reason, we will consider decision tables with only one decision. We adopt the following definition of a decision table:

By a decision table we mean an information system $S = (U, A_1 \cup A_2 \cup \{d\})$, where $d \notin A_1 \cup A_2$ is a distinguished attribute called decision. The elements of A_1 are called stable conditions, whereas the elements of $A_2 \cup \{d\}$ are called flexible conditions. Our goal is to suggest changes in values of attributes in A_1 for some objects from U so the values of the attribute d for these objects may change as well. A formal expression describing such a property is called an action rule [9], [13].

In order to construct action rules, first we need to extract classification rules describing relationships between attributes from A_1 and the attribute d . By $Dom(r)$ we mean all attributes listed in the *IF* part of a classification rule r . For example, if $r = [(a_1, 3) \wedge (a_2, 4) \rightarrow (d, 3)]$ is a rule, then $Dom(r) = \{a_1, a_2\}$. By $d(r)$ we

denote the decision value of rule r . In our example $d(r) = 3$. If r_1, r_2 are rules and $B \subseteq A_1 \cup A_2$ is a set of attributes, then $r_1/B = r_2/B$ means that the conditional parts of rules r_1, r_2 restricted to attributes B are the same. For example if $r_1 = [(a_1, 3) \rightarrow (d, 3)]$, then $r_1/a_1 = r/a_1$. Assume also that $(a, v \rightarrow w)$ denotes the fact that the value of attribute a needs to be changed from v to w for some objects in U . Similarly, the term $(a, v \rightarrow w)(x)$ means that $a(x) = v$ needs to be changed to $a(x) = w$. Saying another words, the property (a, v) of an object x has to be changed to property (a, w) . Assume now that rules r_1, r_2 are extracted from S and $r_1/A_1 = r_2/A_1$, $d(r_1) = k_1$, $d(r_2) = k_2$ and $k_1 < k_2$ (k_1 is ranked lower than k_2). Also, assume that $(b_1, b_2, \dots, b_p)$ is a list of all attributes in $Dom(r_1) \cap Dom(r_2) \cap A_2$ on which r_1, r_2 differ and $r_1(b_1) = v_1$, $r_1(b_2) = v_2, \dots$, $r_1(b_p) = v_p$, $r_2(b_1) = w_1$, $r_2(b_2) = w_2, \dots$, $r_2(b_p) = w_p$. By (r_1, r_2) -action rule on $x \in U$ we mean an expression: $r = [(b_1, v_1 \rightarrow w_1) \wedge (b_2, v_2 \rightarrow w_2) \wedge \dots \wedge (b_p, v_p \rightarrow w_p)](x) \rightarrow (d, k_1 \rightarrow k_2)(x)$. By the support of action rule r we mean the set of all objects in S satisfying the description $[(b_1, v_1) \wedge (b_2, v_2) \wedge \dots \wedge (b_p, v_p) \wedge (d, k_1)]$. Assume now that by $r(x)$ we mean a new object obtained from x by changing its property (b_i, v_i) to (b_i, w_i) for each $1 \leq i \leq p$. If $d(r(x)) = k_2$, then x supports r .

Table 1. Two classification rules extracted from S

<i>Stable</i>	<i>Flexible</i>	<i>Stable</i>	<i>Flexible</i>	<i>Stable</i>	<i>Flexible</i>	<i>Decision</i>
<i>A</i>	<i>B</i>	<i>C</i>	<i>E</i>	<i>G</i>	<i>H</i>	<i>D</i>
a_1	b_1	c_1	e_1			d_1
a_1	b_2			g_2	h_2	d_2

To construct an extended action rule [13], let us assume that two classification rules, each one referring to a different decision class, are considered. We present them in Table 1 to better clarify the process. Here, "*Stable*" means stable classification attribute and "*Flexible*" means flexible one. In a standard representation, these two classification rules have a form:

$$r_1 = [(a_1 \wedge b_1 \wedge c_1 \wedge e_1) \rightarrow d_1], r_2 = [(a_1 \wedge b_2 \wedge g_2 \wedge h_2) \rightarrow d_2].$$

Assume now that object x supports rule r_1 which means that it is classified as d_1 . In order to reclassify x to a higher level class d_2 , we need to change not only the value of B from b_1 to b_2 but also to assume that $G(x) = g_2$ and that the value H for object x has to be changed to h_2 . This is the meaning of the extended (r_1, r_2) -action rule defined by the expression below:

$$r = [(a_1 \wedge (B, b_1 \rightarrow b_2) \wedge g_2 \wedge (H, \rightarrow h_2)](x) \rightarrow (D, d_1 \rightarrow d_2)(x).$$

Assume now that by $Sup(t)$ we mean the number of tuples having property t . By the support of the extended (r_1, r_2) -action rule (given above) we mean:

$Sup[a_1 \wedge b_1 \wedge g_2 \wedge d_1]$. By the confidence of the extended (r_1, r_2) -action rule r (given above) we mean (see [12], [13]):

$$[Sup[a_1 \wedge b_1 \wedge g_2 \wedge d_1] / Sup[a_1 \wedge b_1 \wedge g_2]] \cdot [Sup[a_1 \wedge b_2 \wedge c_1 \wedge d_2] / Sup[a_1 \wedge b_2 \wedge c_1]].$$

Now, we explain the steps needed to compute the confidence of an extended (r_1, r_2) -action rule r . In the first conjunct of the definition of an action rule we are looking for objects described by values of stable attributes only taken from the rule r_2 . Its second conjunct refers to the percentage of objects $r(x)$ which also have a property d_2 , where x supports the first conjunct. Values of stable attributes listed in r_1 do not have to be considered at all. To give another example of an action rule and an extended action rule, assume that $S = (U, A_1 \cup A_2 \cup \{d\})$ is a decision system represented by Table 2. Assume that $A_1 = \{c, b\}$, $A_2 = \{a\}$.

Table 2. Decision System S

X	c	a	b	d
x_1	2	1	1	L
x_2	1	2	2	L
x_3	2	2	1	H
x_4	1	1	1	L

For instance, rules $r_1 = [(a, 1) \wedge (b, 1) \rightarrow (d, L)]$, $r_2 = [(c, 2) \wedge (a, 2) \rightarrow (d, H)]$ can be extracted from S , where $Dom(r_1) = \{x_1, x_4\}$. Extended (r_1, r_2) -action rule $[(a, 1 \rightarrow 2) \wedge (c, 2)](x) \rightarrow [(d, L \rightarrow H)](x)$ is only supported by object x_1 . The corresponding (r_1, r_2) -action rule $[(a, 1 \rightarrow 2)](x) \rightarrow [(d, L \rightarrow H)](x)$ is supported by x_1 and x_4 . The confidence of an extended action rule is higher than the confidence of the corresponding action rule because the object making the confidence of that action rule lower has been removed from its set of support.

3 Action Rules Discovery, the New Approach

Let us assume again that $S = (U, A_1 \cup A_2 \cup \{d\})$ is a decision system, where $d \notin A_1 \cup A_2$ is a distinguished attribute called the decision. The elements of A_1 are called stable conditions, whereas the elements of $A_2 \cup \{d\}$ are called flexible. Assume that $d_1 \in V_d$ and $x \in U$. We say that x is a d_1 -object if $d(x) = d_1$. We also assume that $\{a_1, a_2, \dots, a_p\} \subseteq A_1$, $\{b_1, b_2, \dots, b_q\} \subseteq A_2$, $a_{i,j}$ denotes a value of attribute a_i , $b_{i,j}$ denotes a value of attribute b_i , for any i, j and that

$$r = [[a_{1,1} \wedge a_{2,1} \wedge \dots \wedge a_{p,1} \wedge b_{1,1} \wedge b_{2,1} \wedge \dots \wedge b_{q,1}] \longrightarrow d_1]$$

is a classification rule extracted from S supporting some d_1 -objects in S . By $sup(r)$ and $conf(r)$, we mean the support and the confidence of r , respectively.

Class d_1 is a preferable class and our goal is to reclassify d_2 -objects into d_1 class, where $d_2 \in V_d$.

By an action rule $r_{[d_2 \rightarrow d_1]}$ associated with r and the above reclassification task $(d, d_2 \rightarrow d_1)$ we mean the following expression:

$$r_{[d_2 \rightarrow d_1]} = [[a_{1,1} \wedge a_{2,1} \wedge \dots \wedge a_{p,1} \wedge (b_1, \rightarrow b_{1,1}) \wedge (b_2, \rightarrow b_{2,1}) \wedge \dots \wedge (b_q, \rightarrow b_{q,1})] \rightarrow (d, d_2 \rightarrow d_1)]$$

In a similar way, by an action rule $r_{[\rightarrow d_1]}$ associated with r and the reclassification task $(d, \rightarrow d_1)$ we mean the following expression:

$$r_{[\rightarrow d_1]} = [[a_{1,1} \wedge a_{2,1} \wedge \dots \wedge a_{p,1} \wedge (b_1, \rightarrow b_{1,1}) \wedge (b_2, \rightarrow b_{2,1}) \wedge \dots \wedge (b_q, \rightarrow b_{q,1})] \rightarrow (d, \rightarrow d_1)]$$

The term $[a_{1,1} \wedge a_{2,1} \wedge \dots \wedge a_{p,1}]$, built from values of stable attributes, is called the header of $r_{[d_2 \rightarrow d_1]}$ and its values can not be changed.

The support set of the action rule $r_{[d_2 \rightarrow d_1]}$ is defined as $Sup(r_{[d_2 \rightarrow d_1]}) = \{x \in U : (a_1(x) = a_{1,1}) \wedge (a_2(x) = a_{2,1}) \wedge \dots \wedge (a_p(x) = a_{p,1}) \wedge (d(x) = d_2)\}$.

Clearly, if $conf(r) \neq 1$, then some objects in S satisfying the description $[a_{1,1} \wedge a_{2,1} \wedge \dots \wedge a_{p,1} \wedge b_{1,1} \wedge b_{2,1} \wedge \dots \wedge b_{q,1}]$ are classified as d_2 . According to the rule $r_{[d_2 \rightarrow d_1]}$ they should be classified as d_1 which means that the confidence of $r_{[d_2 \rightarrow d_1]}$ will get decreased.

If $Sup(r_{[d_2 \rightarrow d_1]}) = \emptyset$, then $r_{[d_2 \rightarrow d_1]}$ can not be used for reclassification of objects. Similarly, $r_{[\rightarrow d_1]}$ can not be used for reclassification, if $Sup(r_{[\rightarrow d_1]}) = \emptyset$, for each d_2 where $d_2 \neq d_1$. From the point of view of actionability, such rules are not interesting.

Let $Sup(r_{[\rightarrow d_1]}) = \bigcup \{Sup(r_{[d_2 \rightarrow d_1]}) : (d_2 \in V_d) \wedge (d_2 \neq d_1)\}$ and $Sup(R_{[\rightarrow d_1]}) = \bigcup \{Sup(r_{[\rightarrow d_1]}) : r \in R(d_1)\}$, where $R(d_1)$ is the set of all classification rules extracted from S which are defining d_1 . So, $Sup(R_S) = \bigcup \{Sup(R_{[\rightarrow d_1]}) : d_1 \in V_d\}$ contains all objects in S which potentially can be reclassified.

Assume now that $U(d_1) = \{x \in U : d(x) \neq d_1\}$. Objects in the set $B(d_1) = [U(d_1) - Sup(R_{[\rightarrow d_1]})]$ can not be reclassified to the class d_1 and they are called d_1 -resistant.

Let $B(\neg d_1) = \bigcap \{B(d_i) : (d_i \in V_d) \wedge (d_i \neq d_1)\}$. Clearly $B(\neg d_1)$ represents the set of d_1 -objects which can not be reclassified. They are called d_1 -stable. Similarly, the set $B_d = \bigcup \{B(\neg d_i) : d_i \in V_d\}$ represents objects in U which can not be reclassified to any decision class. All these objects are called d -stable. In order to show how to find them, the notion of a confidence of an action rule is needed.

Let $r_{[d_2 \rightarrow d_1]}$, $r'_{[d_2 \rightarrow d_3]}$ are two action rules extracted from S . We say that these rules are p-equivalent ($\simeq$), if the condition given below holds for every $b_i \in A_1 \cup A_2$:

if r/b_i , r'/b_i are both defined, then $r/b_i = r'/b_i$.

Now, we explain how to calculate the confidence of $r_{[d_2 \rightarrow d_1]}$. Let us take d_2 -object $x \in Sup(r_{[d_2 \rightarrow d_1]})$. We say that x positively supports $r_{[d_2 \rightarrow d_1]}$ if there is no classification rule r' extracted from S and describing $d_3 \in V_d$, $d_3 \neq d_1$, which is p-equivalent to r , such that $x \in Sup(r'_{[d_2 \rightarrow d_3]})$. The corresponding

subset of $Sup(r_{[d_2 \rightarrow d_1]})$ is denoted by $Sup^+(r_{[d_2 \rightarrow d_1]})$. Otherwise, we say that x negatively supports $r_{[d_2 \rightarrow d_1]}$. The corresponding subset of $Sup(r_{[d_2 \rightarrow d_1]})$ is denoted by $Sup^-(r_{[d_2 \rightarrow d_1]})$.

By the confidence of $r_{[d_2 \rightarrow d_1]}$ in S we mean:

$$Conf(r_{[d_2 \rightarrow d_1]}) = [card[Sup^+(r_{[d_2 \rightarrow d_1]})]/card[Sup(r_{[d_2 \rightarrow d_1]})]] \cdot conf(r).$$

Now, if we assume that $Sup^+(r_{[\neg d_1]}) = \bigcup \{Sup^+(r_{[d_2 \rightarrow d_1]}) : (d_2 \in V_d) \wedge (d_2 \neq d_1)\}$, then by the confidence of $r_{[\neg d_1]}$ in S we mean:

$$Conf(r_{[\neg d_1]}) = [card[Sup^+(r_{[\neg d_1]})]/card[Sup(r_{[\neg d_1]})]] \cdot conf(r).$$

Now, we go back to the problem of finding all d -stable objects in S . First of all, we observe that $B^+(d_1) = [U(d_1) - Sup^+(R_{[\neg d_1]})]$ contains all d_1 -resistant objects. Let $B^+(\neg d_1) = \bigcap \{B^+(d_i) : (d_i \in V_d) \wedge (d_i \neq d_1)\}$. Clearly $B^+(\neg d_1)$ is the set of all d_1 -stable objects in S . The same $B_d^+ = \bigcup \{B^+(\neg d_i) : d_i \in V_d\}$ contains all d -stable objects in S .

It can be easily proved that the definition of support and confidence of action rules given in Section 3 is equivalent to the definition of support and confidence given in Section 2.

4 Class-Movability of Objects

In this section we introduce the notion of a class-movability index (m-index) assigned to an object and next we partition objects in U with respect to that index. The m-index associated with an object, shows a rank of the object. Objects of higher rank are seen as objects more preferably movable between decision classes than objects of lower rank. Let $(V_d, \preceq)$ is an ordered set of values of the decision attribute d , where $V_d = \{d_1, d_2, \dots, d_k\}$. Before defining $\preceq$, we introduce function F_S , called *decision attribute ranking* associated with S . It assigns an integer number to each d_i , $1 \leq i \leq k$. If $F_S(d_i) \leq F_S(d_j)$, then $d_i \preceq d_j$.

Now, for each $j \in \{1, 2, \dots, k\}$, we start with a collection of sets $P_j(i) = Sup^+(r_{[d_j \rightarrow d_i]})$, each containing all positively supported d_j -objects in S . It means $P_j(i)$ contains all objects in U which can be reclassified (moved) from the decision class d_j to the class d_i . Let $P_j(N) = \bigcap \{P_j(i) : i \in N\}$, for any $N \subseteq \{1, 2, \dots, k\}$. Clearly, $P_j(N)$ contains all objects in U which can be moved from the decision class d_j to any of the classes d_i , where $i \in N$.

Before we give the definition of m-index assigned to an object, we introduce the notion of m-index assigned to N_j , where $N_j \subseteq \{1, 2, \dots, k\}$. Let $N_j^+ = \{i \in N : F_S(d_i) - F_S(d_j) > 0\}$, for any $N \subseteq \{1, 2, \dots, k\}$. The set N_j^+ represents all attribute values in $\{d_i : i \in N\}$ which have higher decision attribute ranking than d_j .

By class-movability index (m-index) assigned to N_j , we mean: $ind(N_j) = \sum \{F_S(d_i) - F_S(d_j) : i \in N_j^+\}$.

By class-movability index (m-index) assigned to d_j -object x , we mean: $ind_S(x) = \max \{ind(N_j) : N_j \subseteq \{1, 2, \dots, k\} \wedge x \in P_j(N)\}$.

In general, $ind_S(x)$ shows the overall amount of improvement open to x in terms of a number of available re-classifications and the improvements linked with every re-classification.

5 An Example

Let us assume that the decision system $S = (U, \{A_1 \cup A_2 \cup \{d\}\})$, where $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$, is represented by Table 3. A number of different methods can be used to extract rules in which the THEN part consists of the decision attribute d and the IF part consists of attributes belonging to $A_1 \cup A_2$. In our example, the set $A_1 = \{a, b, c\}$ contains stable attributes and $A_2 = \{e, f, g\}$ contains flexible attributes. System *LERS* [4] is used in this paper for rules extraction.

Table 3. Decision System

U	a	b	c	e	f	g	g
x_1	a_1	b_1	c_1	e_1	f_2	g_1	d_1
x_2	a_2	b_1	c_2	e_2	f_2	g_2	d_3
x_3	a_3	b_1	c_1	e_2	f_2	g_3	d_2
x_4	a_1	b_1	c_2	e_2	f_2	g_1	d_2
x_5	a_1	b_2	c_1	e_3	f_2	g_1	d_2
x_6	a_2	b_1	c_1	e_2	f_3	g_1	d_2
x_7	a_2	b_3	c_2	e_2	f_2	g_2	d_2
x_8	a_2	b_1	c_1	e_3	f_2	g_3	d_2

We are interested in reclassifying d_2 -objects either to class d_1 or d_3 . Four certain classification rules describing d_1, d_3 can be extracted by *LERS* from the decision system S . They are given below:

$r1 = [b_1 \wedge c_1 \wedge f_2 \wedge g_1] \rightarrow d_1$, $r2 = [a_2 \wedge b_1 \wedge e_2 \wedge f_2] \rightarrow d_3$, $r3 = e_1 \rightarrow d_1$, $r4 = [b_1 \wedge g_2] \rightarrow d_3$.

It can be checked that $R_{[d, \rightarrow d_1]} = \{r1, r3\}$ and $R_{[d, \rightarrow d_3]} = \{r2, r4\}$. Action rules associated with $r1, r2, r3, r4$ and the reclassification task either $(d, d_2 \rightarrow d_1)$ or $(d, d_2 \rightarrow d_3)$ are:

$$\begin{aligned}
 r1_{[d_2 \rightarrow d_1]} &= [b_1 \wedge c_1 \wedge (f, \rightarrow f_2) \wedge (g, \rightarrow g_1)] \rightarrow (d, d_2 \rightarrow d_1), \\
 r2_{[d_2 \rightarrow d_3]} &= [a_2 \wedge b_1 \wedge (e, \rightarrow e_2) \wedge (f, \rightarrow f_2)] \rightarrow (d, d_2 \rightarrow d_3), \\
 r3_{[d_2 \rightarrow d_1]} &= [(e, \rightarrow e_1)] \rightarrow (d, d_2 \rightarrow d_1), \\
 r4_{[d_2 \rightarrow d_3]} &= [b_1 \wedge (g, \rightarrow g_2)] \rightarrow (d, d_2 \rightarrow d_3).
 \end{aligned}$$

It can be easily shown that $Sup(r1_{[d_2 \rightarrow d_1]}) = \{x_3, x_6, x_8\}$, $Sup(r2_{[d_2 \rightarrow d_3]}) = \{x_6, x_8\}$, $Sup(r3_{[d_2 \rightarrow d_1]}) = \{x_3, x_4, x_5, x_6, x_7, x_8\}$, $Sup(r4_{[d_2 \rightarrow d_3]}) = \{x_3, x_4, x_6, x_8\}$. The same, $Sup(r1_{[\rightarrow d_1]}) = \{x_3, x_4, x_5, x_6, x_7, x_8\}$, $Sup(r2_{[\rightarrow d_3]}) = \{x_3, x_4, x_6, x_8\}$. So, all d_2 -objects can be potentially reclassified to the class d_1 . On the other hand, d_2 -objects only from the set

$\{x_3, x_4, x_6, x_8\}$ can be potentially reclassified to the same class d_1 . It means that d_2 -objects $\{x_5, x_7\}$ are d_1 -resistant.

Now, let us notice that action rules $r1_{[d_2 \rightarrow d_1]}$, $r2_{[d_2 \rightarrow d_3]}$ are p-equivalent. Since $Sup(r1_{[d_2 \rightarrow d_1]}) = \{x_3, x_6, x_8\}$ and $Sup(r2_{[d_2 \rightarrow d_3]}) = \{x_6, x_8\}$, then only x_3 positively supports $r1_{[d_2 \rightarrow d_1]}$ and objects in $\{x_6, x_8\}$ negatively support both rules $r1_{[d_2 \rightarrow d_1]}$, $r2_{[d_2 \rightarrow d_3]}$. So, the confidence:

$$Conf(r1_{[d_2 \rightarrow d_1]}) = [card[Sup^+(r1_{[d_2 \rightarrow d_1]})]/card[Sup(r1_{[d_2 \rightarrow d_1]})]] \cdot conf(r1) = [1/3] \cdot 1 = 1/3.$$

$$Conf(r2_{[d_2 \rightarrow d_3]}) = [card[Sup^+(r2_{[d_2 \rightarrow d_3]})]/card[Sup(r2_{[d_2 \rightarrow d_3]})]] \cdot conf(r2) = [0/2] \cdot 1 = 0.$$

6 Conclusion

The action rules discovery process, presented in this paper, differs significantly from previous strategies because we use only single classification rules, instead of their pairs, to construct action rules. This difference is quite essential because the number of classification rules discovered from a database is usually large. The previous strategies (see [9], [12], [13], [14]) require to consider all pairs of classification rules, defining different decision values, which makes their time complexity quite high. Also, the proposed new approach to action rules discovery allows to define their confidence in a more natural way than in previous papers (see [12]). The results presented in [3], [8], [12], [13], [14] need to be generalized to match this new definition of an action rule. But, this generalization process should be rather straightforward.

Our initial implementation of the proposed new method for action rules discovery outperforms the previous strategy [12], [13], [14] in terms of time complexity. Clearly, the confidence and support of action rules is strictly dependent on the support and confidence of classification rules used to construct them.

Acknowledgements

This research was partially supported by the National Science Foundation under grant IIS-0414815.

References

1. Adomavicius, G., Tuzhilin, A. (1997), *Discovery of actionable patterns in databases: the action hierarchy approach*, in Proceedings of KDD97 Conference, Newport Beach, CA, AAAI Press
2. Geffner, H., Wainer, J. (1998), *Modeling action, knowledge and control*, ECAI 98, 13th European Conference on AI, (Ed. H. Prade), John Wiley & Sons, 532-536
3. Greco, S., Matarazzo, B., Pappalardo, N., Slowiński, R. (2005), *Measuring expected effects of interventions based on decision rules*, Journal of Experimental and Theoretical Artificial Intelligence, Taylor Francis, Vol. 17, No. 1-2

4. Chmielewski, M.R., Grzymala-Busse, J.W., Peterson, N.W., Than, S. (1993), *The rule induction system LERS - a version for personal computers*, Foundations of Computing and Decision Sciences, Vol. 18, No. 3-4, Institute of Computing Science, Technical University of Poznan, Poland, 181-212
5. Liu, B., Hsu, W., Chen, S. (1997), *Using general impressions to analyze discovered classification rules*, Proceedings of KDD97 Conference, Newport Beach, CA, AAAI Press
6. Pawlak, Z.(1991), *Information systems - theoretical foundations*, Information Systems Journal, Vol. 6, 205-218
7. Raś, Z.W., Gupta, S. (2002), *Global action rules in distributed knowledge systems*, Fundamenta Informaticae Journal, IOS Press, Vol. 51, No. 1-2, 175-184
8. Raś, Z., Wieczorkowska, A. (2000), *Action Rules: how to increase profit of a company*, in Principles of Data Mining and Knowledge Discovery, (Eds. D.A. Zighed, J. Komorowski, J. Zytkow), Proceedings of PKDD'00, Lyon, France, LNAI, No. 1910, Springer-Verlag, 587-592
9. Raś, Z.W., Tzacheva, A., Tsay, L.-S. (2005), *Action rules*, Encyclopedia of Data Warehousing and Mining, (Ed. J. Wang), Idea Group Inc., 1-5
10. Silberschatz, A., Tuzhilin, A., (1995), *On subjective measures of interestingness in knowledge discovery*, Proceedings of KDD'95 Conference, AAAI Press
11. Silberschatz, A., Tuzhilin, A., (1996), *What makes patterns interesting in knowledge discovery systems*, IEEE Transactions on Knowledge and Data Engineering Vol. 5, No. 6
12. Tsay, L.-S., Raś, Z.W. (2005), *Action Rules Discovery System DEAR, Method and Experiments*, Journal of Experimental and Theoretical Artificial Intelligence, Taylor & Francis, Vol. 17, No. 1-2, 119-128
13. Tsay, L.-S., Raś, Z.W., Dardzinska, A., *Mining E-Action Rules*, in *Mining Complex Data*, Proceedings of 2005 IEEE ICDM Workshop in Houston, Texas, Published by Math. Dept., Saint Mary's Univ., Nova Scotia, Canada, 2005, 85-90
14. Tzacheva, A., Raś, Z.W. (2005), *Action rules mining*, International Journal of Intelligent Systems, Wiley, Vol. 20, No. 7, 719-736

Characteristics of Indiscernibility Degree in Rough Clustering Examined Using Perfect Initial Equivalence Relations

Shoji Hirano and Shusaku Tsumoto

Department of Medical Informatics, Shimane University, School of Medicine
89-1 Enya-cho, Izumo, Shimane 693-8501, Japan
hirano@ieee.org, tsumoto@computer.org

Abstract. In this paper, we analyze the influence of the indiscernibility degree, which is a primary parameter in rough clustering, on cluster formation. Rough clustering consists of two steps: (1) assignment of initial equivalence relations and (2) iterative refinement of the initial relations. Indiscernibility degree plays a key role in the second step, but it is not easy to independently analyze its characteristics because it inherits the results of step 1. In this paper, we employ the perfect initial equivalence relations, which were generated according to class labels of data, to seclude the influence of step 1. We first examine the relationship between the threshold value of indiscernibility degree and resultant clusters. After that, we apply random disturbance to the perfect relations, and examine how the result changes. The results demonstrated that the relationships between indiscernibility degree and the number of clusters draw a globally convex but multi-modal curve, and the range of indiscernibility degree that yields best cluster validity may exist on a local minimum around the global one which generates single cluster.

Keywords: Indiscernibility, Clustering, Rough Set.

1 Introduction

In this paper, we analyze the influence of the indiscernibility degree, which is a primary parameter in rough clustering, on cluster formation. Rough clustering [1] is a novel grouping method that controls the granularity of discrimination by a measure called indiscernibility degree. The clustering results can be affected by two factors: (1) initial equivalence relations that form the basic partition of objects, (2) threshold value of indiscernibility degrees. In previous work, we have shown that (1) minor disturbance in the initial equivalence relations would be absorbed in refinement steps [2], and (2) there might be a range of indiscernibility degree that yield high cluster validity. However, these findings were all dependent on the method for determining initial equivalence relations - in this case density-based determination method.

In order to seclude the effect of the method for initial equivalence determination, we employ the perfect initial equivalence relations that are derived from

class labels of objects. Based on the perfect equivalence relations, we first examine the relationship between the threshold value of indiscernibility degree and resultant clusters. After that, we apply random disturbance to the perfect relations, and examine how the result changes. The remainder of this paper is organized as follows. Section 2 gives a brief explanation of rough clustering. Section 3 describes experimental results on artificial datasets, and Section 4 concludes the technical results.

2 Rough Clustering Overview

This section gives a brief overview of rough clustering, AAA, indiscernibility-based clustering. This method is based on iterative refinement of N binary classifications, where N denotes the number of objects. First, an equivalence relation, that classifies all the other objects into two classes, is assigned to each of N objects by referring to the relative proximity. Next, for each pair of objects, the number of binary classifications in which the pair is included in the same class is counted. This number is termed the indiscernibility degree. If the indiscernibility degree of a pair is larger than a user-defined threshold value, the equivalence relations may be modified so that all of the equivalence relations commonly classify the pair into the same class. This process is repeated until class assignment becomes stable. Consequently, we may obtain the clustering result that follows a given level of granularity. The main benefits of this method is that (1) it can handle relative proximity, where no geometric measure such as centroids can not be defined, (2) it can take dissimilarity matrix as input and does not require any direct reference to the original data value.

There are two parameters that control the behavior of this clustering method: the threshold value T_h for refinement of equivalence relations and the number N_r of iteration of refinement. As shown in the experiments, N_r can be determined automatically, because the equivalence relations will be stable after several cycles of refinement. The refinement process can be terminated when no candidates for refinement appear.

2.1 Assignment of Initial Equivalence Relations

Let $U = \{x_1, x_2, \dots, x_N\}$ be the set of objects we are interested in. An equivalence relation R_i for object x_i is defined by

$$U/R_i = \{P_i, U - P_i\}, \quad (1)$$

where

$$P_i = \{x_j \mid d(x_i, x_j) \leq Th_{di}\}, \quad \forall x_j \in U. \quad (2)$$

$d(x_i, x_j)$ denotes dissimilarity between objects x_i and x_j , and Th_{di} denotes an upper threshold value of dissimilarity for object x_i . The equivalence relation, R_i classifies U into two categories: P_i , which contains objects similar to x_i and $U - P_i$, which contains objects dissimilar to x_i . When $d(x_i, x_j)$ is smaller than

Th_{di} , object x_j is considered to be indiscernible to x_i . U/R_i can be alternatively written as $U/R_i = \{\{[x_i]_{R_i}\}, \{\overline{[x_i]_{R_i}}\}\}$, where $[x_i]_{R_i} \cap \overline{[x_i]_{R_i}} = \phi$ and $[x_i]_{R_i} \cup \overline{[x_i]_{R_i}} = U$ hold.

Methods for constructing initial equivalence relations, including the choice of dissimilarity measure, is arbitrary under the condition that it has the ability of performing binary classification of U . For example, one can simply use Euclidean distance and k-means with cluster number 2, if it is appropriate based on the property of the data. We have introduced a method for constructing initial equivalence relations based on the denseness of the objects in [1]; however, one may use another approach for this purpose.

2.2 Refinement of Initial Equivalence Relations

In the second stage, we perform global optimization of initial equivalence relations so that they produce adequately coarse classification to the objects. The global similarity of objects is represented by a newly introduced measure, the *indiscernibility degree*. Rough clustering takes a threshold value of the indiscernibility degree as an input and associates it with the user-defined granularity of the categories. Given the threshold value, we iteratively refine the initial equivalence relations in order to produce categories that meet the given level of granularity.

Now let us assume $U = \{x_1, x_2, x_3, x_4, x_5\}$ and classifications of U by $\mathbf{R} = \{R_1, R_2, R_3, R_4, R_5\}$ is given as follows.

$$\begin{aligned} U/R_1 &= \{\{x_1, x_2, x_3\}, \{x_4, x_5\}\}, \\ U/R_2 &= \{\{x_1, x_2, x_3\}, \{x_4, x_5\}\}, \\ U/R_3 &= \{\{x_2, x_3, x_4\}, \{x_1, x_5\}\}, \\ U/R_4 &= \{\{x_1, x_2, x_3, x_4\}, \{x_5\}\}, \\ U/R_5 &= \{\{x_4, x_5\}, \{x_1, x_2, x_3\}\}. \end{aligned} \quad (3)$$

This example contains three types of equivalence relations: R_1 ($= R_2 = R_5$), R_3 and R_4 . Since each of them classifies U slightly differently, classification of U by the family of equivalence relations $\mathbf{R}$, $U/\mathbf{R}$, contains four very small, almost independent categories.

$$U/\mathbf{R} = \{\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}\}. \quad (4)$$

In the following we present a method to reduce the variety of equivalence relations and to obtain coarser categories.

First, we define an *indiscernibility degree*, $\gamma(x_i, x_j)$, for two objects x_i and x_j as follows.

$$\gamma(x_i, x_j) = \frac{\sum_{k=1}^{|U|} \delta_k^{indis}(x_i, x_j)}{\sum_{k=1}^{|U|} \delta_k^{indis}(x_i, x_j) + \sum_{k=1}^{|U|} \delta_k^{dis}(x_i, x_j)}, \quad (5)$$

where

$$\delta_k^{indis}(x_i, x_j) = \begin{cases} 1, & \text{if } (x_i \in [x_k]_{R_k} \wedge x_j \in [x_k]_{R_k}) \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

and

$$\delta_k^{dis}(x_i, x_j) = \begin{cases} 1, & \text{if } (x_i \in [x_k]_{R_k} \wedge x_j \notin [x_k]_{R_k}) \\ & \text{or if } (x_i \notin [x_k]_{R_k} \wedge x_j \in [x_k]_{R_k}) \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Equation (6) shows that $\delta_k^{indis}(x_i, x_j)$ takes 1 only when the equivalence relation R_k regards both x_i and x_j as indiscernible objects, under the condition that both of them are in the same equivalence class as x_k . Equation (7) shows that $\delta_k^{dis}(x_i, x_j)$ takes 1 only when R_k regards x_i and x_j as discernible objects, under the condition that either of them is in the same class as x_k . By summing $\delta_k^{indis}(x_i, x_j)$ and $\delta_k^{dis}(x_i, x_j)$ for all k ($1 \leq k \leq |U|$) as in Equation (5), we obtain the percentage of equivalence relations that regard x_i and x_j as indiscernible objects. Note that in Equation (6), we excluded the case when x_i and x_j are indiscernible but not in the same class as x_k . This is to exclude the case where R_k does not significantly put weight on discerning x_i and x_j . As mentioned in Section 2.1, P_k for R_k is determined by focusing on similar objects rather than dissimilar objects. This means that when both of x_i and x_j are highly dissimilar to x_k , their dissimilarity is not significant for x_k , when determining the dissimilarity threshold Th_{dk} . Thus we only count the number of equivalence relations that certainly evaluate the dissimilarity of x_i and x_j .

From its definition, a large $\gamma(x_i, x_j)$ represents that x_i and x_j are commonly regarded as indiscernible objects by the large number of the equivalence relations. Therefore, if an equivalence relation R_l discerns the objects that have high γ value, we consider that it represents excessively fine classification knowledge and refine it according to the following procedure (note that R_l is rewritten as R_i below for the purpose of generalization).

Let $R_i \in \mathbf{R}$ be an initial equivalence relation on U . A refined equivalence relation $R'_i \in \mathbf{R}'$ of R_i is defined as

$$U/R'_i = \{P'_i, U - P'_i\}, \quad (8)$$

where P'_i denotes a set of objects represented by

$$P'_i = \{x_j | \gamma(x_i, x_j) \geq Th_h\}, \quad \forall x_j \in U. \quad (9)$$

and Th_h denotes the lower threshold value of the indiscernibility degree above, in which x_i and x_j are regarded as indiscernible objects. It represents that when $\gamma(x_i, x_j)$ is larger than Th_h , R_i is modified to include x_j into the class of x_i .

2.3 Iterative Refinement of Equivalence Relations

It should be noted that the state of the indiscernibility degrees could also be changed after refinement of the equivalence relations, since the degrees are recalculated using the refined family of equivalence relations $\mathbf{R}'$. Thus we iterate the refinement process using the same Th_h until the categories become stable. Note that each refinement process is performed using the previously 'refined' set of equivalence relations.

Table 1. Number of data points in datasets

Dataset	CBS 1	CBS 2	CBS 3	CBS 4	CBS 5	total
c3-1	52	40	93	—	—	185
c3-2	224	31	177	—	—	432
c5-1	52	171	148	215	55	641
c5-2	64	164	126	58	155	567

3 Experimental Results

3.1 Perfect Equivalence Relations

Our aim is to analyze the relationships between the threshold value T_h of indiscernibility degree γ and cluster numbers, while minimizing the influence of methods for determining initial equivalence relations in step 1. We prepared equivalence relations called perfect equivalence relations, which can classify the data into correct groups. Taking them as initial equivalence relations, we performed step 2 of the rough clustering several times by changing T_h and observed the change of resultant clusters. We also performed clustering experiments on randomly disturbed perfect equivalence relations.

A perfect equivalence relation R_i for object x_i is denoted as follows.

$$U/R_i = \{P_i, U - P_i\}, \quad (10)$$

where

$$P_i = \{x_j \mid c[x_i] = c[x_j]\}, \quad \forall x_j \in U. \quad (11)$$

where $c[x_i]$ denotes the class label of x_i assigned when creating the dataset. Obviously, the types of perfect equivalence relations in $\mathbf{R}$ are equal to the number of classes in the dataset, because if objects x_i and x_j belong to the same class, R_i and R_j become identical.

3.2 Datasets

We artificially created a total of four numerical datasets named c3-1, c3-2, c5-1, and c5-2 shown in Table 1. Datasets c3-1 and c3-2 contain three clusters, and c5-1 and c5-2 contain five clusters respectively. The number of data points in each cluster was controlled to be substantially different and in balanced, because the balanced data may induce special effect of T_h on a specific range. The data points were generated based on a two-dimensional normal distribution for easy visualization; however, note that the geometric distribution of data points is not significant in this experiment because we used only their class labels for creating the perfect equivalence relations.

3.3 Procedures

The following procedure was applied to each dataset.

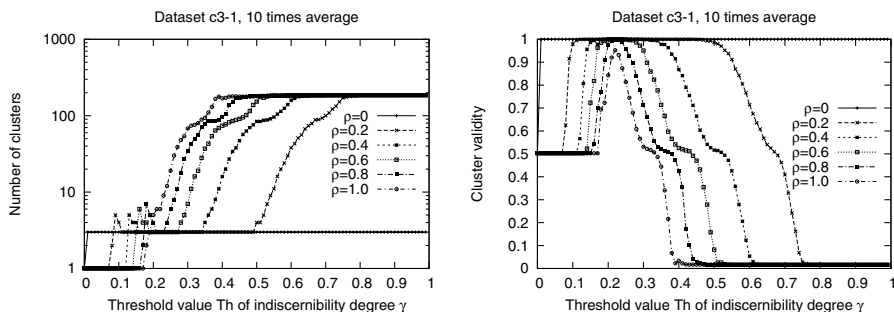


Fig. 1. Results for Dataset c3-1. Left: Number of clusters. Right: Cluster Validity.

1. Form perfect initial equivalence relations: according to E's. (10) and (11), assign a perfect initial equivalence relation for each $x_i \in U$.
2. Disturb the initial relations: Select one of the following disturbance operation randomly at each time and apply it to initial relations. This process is repeated $\text{card}(P_i) \times \rho$ times, where ρ denotes disturbance ratio (from 0.0 to 1.0, interval 0.2).

Delete: Randomly select one element in P_i and remove it from P_i .

Add: Randomly select one element from U and add it to P_i .

Replace: Randomly select one element from P_i and replace it with randomly selected element in U .

3. Clustering: Apply the iterative refinement process of rough clustering to the disturbed initial equivalence relations and obtain clusters. This process is repeated by changing T_h (from 0 to 1.0, interval 0.05). For each T_h , calculate the validity of clustering result according to the following measure.

$$v_{\mathbf{R}}(C) = \min \left(\frac{|X_{\mathbf{R}} \cap C|}{|X_{\mathbf{R}}|}, \frac{|X_{\mathbf{R}} \cap C|}{|C|} \right),$$

where $X_{\mathbf{R}}$ and C denote the obtained clusters and original classes, respectively.

3.4 Results and Discussions

Figures 1-4 show the results on the four datasets respectively. Each of the figures consists of two sub-figures: Th-Number of clusters curves (left) and Th-Cluster Validity curves (right). The horizontal axis corresponds to the threshold value T_h of indiscernibility degree γ . The vertical axis corresponds to the number of clusters or cluster validity for the left or right figure, respectively. Each figure contains six curves indexed by " $\rho = x$ ", which corresponds to the ratio of disturbance of the perfect initial equivalence relations described previously.

Let us first see the global characteristics the curves. At $T_h = 0$, every equivalence relation was modified to include all objects. This means that, regardless of the characteristics of initial equivalence relations, all objects would be grouped into the same cluster. Therefore the number of clusters was always 1 at $T_h = 0$.

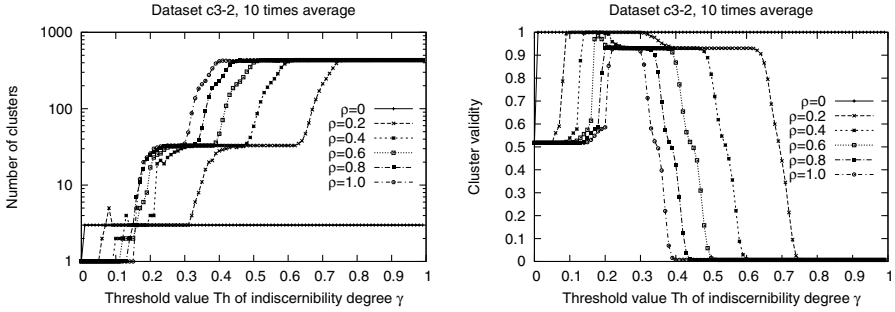


Fig. 2. Results for Dataset c3-2. Left: Number of clusters. Right: Cluster Validity.

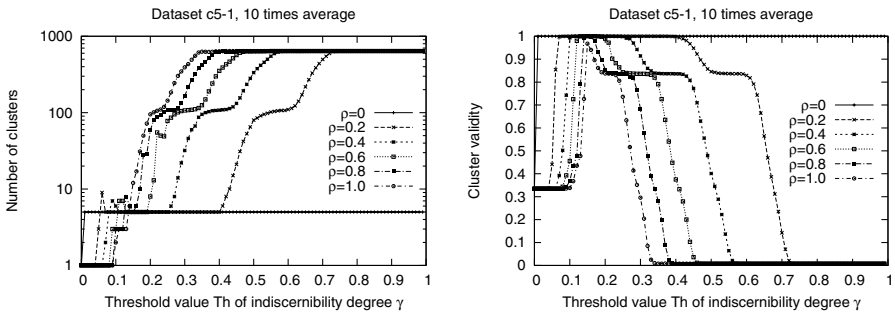


Fig. 3. Results for Dataset c5-1. Left: Number of clusters. Right: Cluster Validity.

The cluster validity took a constant value which was dependent only to the class distribution of the dataset (around 0.5 or 0.3 for the datasets used here).

When $\rho = 0$, initial equivalence relations were identical to the perfect relations since no disturbance was applied. In this case the indiscernibility degrees were 0 for all pairs of objects belonging to different clusters, and 1 for those belonging to the same cluster. Therefore, correct clusters of validity=1 were formed for all values of $T_h > 0$ without any refinement.

If $\rho > 0$, situations become close to those of real-world datasets. The variety of initial equivalence relations drastically increase because of disturbance. Even a small difference of equivalence relations results in producing fine clusters due to the increase of total discrimination ability. Hence, without refinement of the relations, excessively large number of fine clusters would be produced. Let us first see the case of dataset c3-1 in Figure 1. For large values of $T_h > 0.8$, only a few equivalence relations satisfied the condition for refinement in CEQ. (9). As most of the relations remained unchanged, the number of induced clusters kept high value - almost equal to the number of objects in the dataset. When T_h became smaller, the number of equivalence relations to be refined increased. The refinement made classification coarser and made the number of clusters smaller, inducing the increase of cluster validity. The level of T_h for starting this

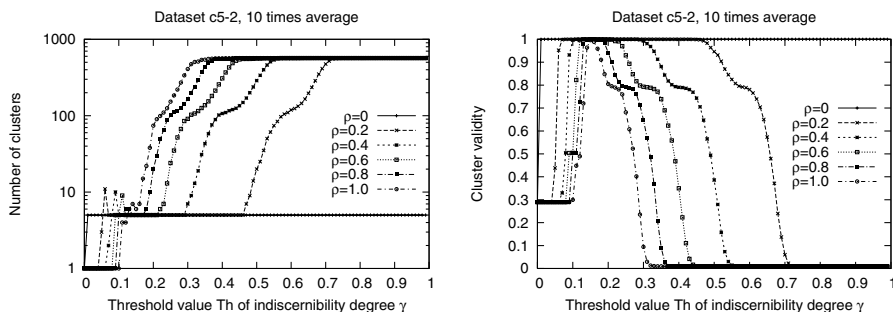


Fig. 4. Results for Dataset c5-2. Left: Number of clusters. Right: Cluster Validity.

improvement was higher if ρ was smaller, because at smaller ρ initial equivalence relations were only slightly and locally modified from the perfect equivalence. Therefore, the indiscernibility degree of each object pairs kept high value, while the types of equivalence relations are quite large. As ρ becomes larger, more severe and global disturbance could occur. Since it induced the decrease of average level of the indiscernibility, the values of T_h should be smaller to do the necessary refinement.

For $0.5 > T_h > 0.1$, the number of clusters kept 3 with the highest validity of 1. In this range, the method could produce the correct cluster assignment with the help of iterative refinement of the disturbed initial equivalence relations. The range became narrow as ρ became small. For example, when $\rho = 0.6$ the range was about $2.7 > T_h > 1.8$ and when $\rho = 1.0$, there was no range of T_h that could generate the correct cluster assignment. If there exists too much disturbance, the level of indiscernibility degrees for objects that should belong to the same cluster would be close to those of objects that belong to different clusters. Hence it would be difficult to form correct clusters, especially for small clusters.

For small values of $T_h < 0.1$, the number of clusters decreased to 1, followed by the decrease of cluster validity. In this range, too coarse cluster was obtained due to too much refinement of equivalence relations. Let us denote by $\min_\gamma$ the minimum value of indiscernibility degrees. Actually, for $Th < \min_\gamma$, the results are identical with the case of $T_h = 0$, due to the discrete property of indiscernibility degree.

The above characteristics were commonly observed for all the other datasets used in this experiment. It demonstrated that, by changing the threshold value of indiscernibility degree, we could control the roughness of classification knowledge, namely, granularity of the data.

Furthermore, an interesting feature about the number of clusters was observed on all datasets. Around $T_h = 0.1 - 0.2$, there existed a short spike at the left end of the range for yielding the correct number of clusters. Although it could disappears on extremely disturbed cases, the convex features of the curve may be used for determining the best range of T_h semi-automatically.

4 Conclusions

In this work, we have empirically investigated the characteristics of indiscernibility degree in rough clustering. By the use of perfect equivalence relations, we could observe more basic relationships between the threshold value of indiscernibility degree and resultant clusters, without the effect of methods for determining initial equivalence relations. The result demonstrated that the threshold parameter might be associated with roughness of knowledge, which also controls the granularity of dataset. Additionally, although it still requires exploratory approach, the convex shape of th-AC curve suggested the possibility of guiding appropriate range of the thresholds. It remains as a future work to investigate the reason why these spikes occur. The future work also include comparison with other methods, e.g, classical hierarchical and partitional clustering methods [3] and rough set-based clustering methods [4].

References

1. Hirano, S., Tsumoto, S.: An indiscernibility-based clustering method with iterative refinement of equivalence relations - rough clustering -. *Journal of Advanced Computational Intelligence and Intelligent Informatics* **7** (2003) 169–177
2. Hirano, S., Tsumoto, S.: On Constructing Clusters from Non-Euclidean Dissimilarity Matrix by Using Rough Clustering. *Lecture Notes in Artificial Intelligence*, **4012** (2006) 5-16, Springer.
3. Everitt B. S., Landau S., Leese M. (2001): *Cluster Analysis* Fourth Edition. Arnold Publishers.
4. Lingras P., Rough set clustering for web mining. *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems* (2002) 1039–1044.

Implication Strength of Classification Rules

Gilbert Ritschard¹ and Djamel A. Zighed²

¹ Dept of Econometrics, University of Geneva, CH-1211 Geneva 4, Switzerland
`gilbert.ritschard@themes.unige.ch`

² Laboratoire ERIC, University of Lyon 2, C.P.11 F-69676 Bron Cedex, France
`abdelkader.zighed@univ-lyon2.fr`

Abstract. This paper highlights the interest of implicative statistics for classification trees. We start by showing how Gras' implication index may be defined for the rules derived from an induced decision tree. Then, we show that residuals used in the modeling of contingency tables provide interesting alternatives to Gras' index. We then consider two main usages of these indexes. The first is purely descriptive and concerns the a posteriori individual evaluation of the classification rules. The second usage, considered for instance by Zighed and Rakotomalala [15], relies upon the intensity of implication to define the conclusion in each leaf of the induced tree.

1 Introduction

Implicative statistics has been introduced by the French mathematician Gras [6, 8, 9] as a tool for data analysis and has, more recently, been exploited for deriving valuable interestingness measures for association rules of the form “If A is observed, then we are very likely to observe B too” [3, 5, 10, 14]. The basic idea behind implicative statistics is that the fewer counter-examples a statistically observed relationship admits, the more implicative it is. It states also that a rule is irrelevant when the observed number of counter-examples exceeds the number expected in case of independence between the premise and the conclusion. Though, as we will show, this concept of strength of implication is applicable in a straightforward manner to classification rules for instance, only little attention has been paid to this appealing idea in the framework of supervised learning.

The aim of this paper is to discuss the scope and limits of implicative statistics for supervised classification and especially for classification trees. Section 2 shows how Gras' implication indexes can be applied to classification rules derived from an induced decision tree. It proposes alternatives to Gras' index inspired from residuals used in the modeling of multiway contingency tables. Section 3 discusses the use of implication strength for the individual validation of each classification rule, while Section 4 shows that the implication strength provides a useful alternative to the majority rule for selecting the most relevant conclusion in a leaf. Section 5 proposes concluding remarks and perspectives of development.

2 Classification Trees and Implication Indexes

For our discussion, we consider a fictional example where we are interested in predicting the civil status (married, single, divorced/widowed) of individuals from their sex (male, female) and sector of activity (primary, secondary, tertiary). The civil status is the outcome or response variable, while sex and activity sector are the predictors. The data set is composed of the 273 cases described by Table 1.

Table 1. The illustrative data set

Civil status	Sex	Activity sector	Number of cases
married	male	primary	50
married	male	secondary	40
married	male	tertiary	6
married	female	primary	0
married	female	secondary	14
married	female	tertiary	10
single	male	primary	5
single	male	secondary	5
single	male	tertiary	12
single	female	primary	50
single	female	secondary	30
single	female	tertiary	18
divorced/widowed	male	primary	5
divorced/widowed	male	secondary	8
divorced/widowed	male	tertiary	10
divorced/widowed	female	primary	6
divorced/widowed	female	secondary	2
divorced/widowed	female	tertiary	2

2.1 Trees and Rules

Classification trees induce classification rules from data in two steps. First, the tree is grown by seeking, through recursive splits of the learning data set, some optimal partition of the predictor space for predicting the outcome class. Each split is done according to the values of one predictor. The process is greedy. It starts by trying all predictors to find the “best” split of the whole learning data set. Then, the process is repeated at each new node until some stopping rule is reached. In a second step, once the tree is grown, classification rules are derived by choosing the most relevant value, usually the majority class, in each leaf (terminal node) of the tree.

Figure 1 shows the tree induced with the CHAID method [11], a 5% significance level and a minimal node size fixed to 20. The same tree is obtained with CART [4] and a minimal .02 gain value. The tree partitions the predictor space into groups such that the distribution of the outcome variable, the civil status, differs as much as possible from one group to the other. For our discussion, it

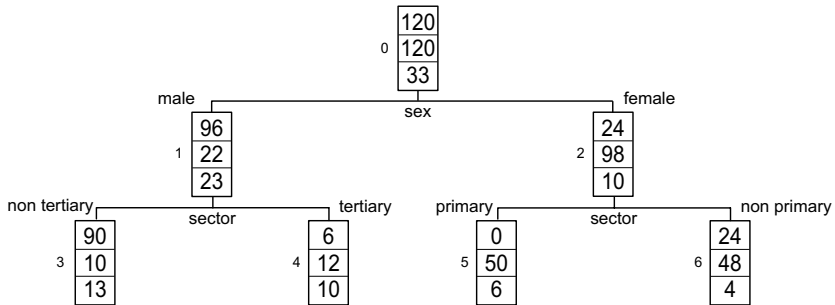


Fig. 1. Example: Induced tree for civil status (married, single, divorced/widowed)

is convenient to represent the four resulting distributions into a table that cross classifies the outcome variable with the set of profiles (the premises of the rules) defined by the branches. Table 2 is thus associated to the tree of Figure 1.

Classification rules are usually derived from the tree by assigning the majority class of the leaf to the branch that leads to it. For example, a man working in the secondary sector belongs to leaf 3 and will be classified as married, while a man of the tertiary sector (leaf 4) will be classified as single. In Table 2, the column headings define the premises of the rules, the conclusion being given, for each column, by the row containing the greatest count. The tree defines thus the four following rules:

- R1: Man of primary or secondary sector $\Rightarrow$ married
- R2: Man of tertiary sector $\Rightarrow$ single
- R3: Woman of primary sector $\Rightarrow$ single
- R2: Woman of secondary or tertiary sector $\Rightarrow$ single

Classification rules as compared with association rules have the following characteristics: i) Their conclusions can only be values (classes) of the outcome variable, and ii) the premises of the rules are mutually exclusive and define a partition of the predictor space. Anyway, they are rules and we can then apply to them concepts such as support, confidence and, which is here our concern, implication indexes.

Table 2. Table associated to the induced tree

Civil Status	Man		Woman		Total
	primary or secondary	tertiary	primary	secondary or tertiary	
Married	90	6	0	24	120
Single	10	12	50	48	120
Div./Widowed	13	10	6	4	33
Total	113	28	56	76	273

2.2 Counter-Examples and Implication Index

The index of implication [see for instance 7, p. 19] of a rule is defined from the number of counter-examples, i.e. of cases that verify the premise but not the conclusion. In our case, it is in each leaf (column of Table 2) the number of cases that are not in the majority class. Letting b denote the conclusion (row of the table) of rule j and n_{bj} the maximum in the j th column, the number of counter-examples is $n_{\bar{b}j} = n_{\cdot j} - n_{bj}$. The index of implication is a standardized form of the deviation between this number and the number of counter-examples expected when assuming that the distribution of the outcome values is independent of the premise.

Formally, the independence hypothesis H_0 states that the number $N_{\bar{b}j}$ of counter-examples of rule j results from a random draw of $n_{\cdot j}$ cases. Under H_0 , letting $n_{b\cdot}/n$ be the marginal proportion of cases in the conclusion class b of rule j , $N_{\bar{b}j}$ follows a binomial distribution $\text{Bin}(n_{\cdot j}, n_{b\cdot}/n)$, or, when $n_{\cdot j}$ is not fixed a priori, a Poisson distribution with parameter $n_{\bar{b}j}^e = n_{\bar{b}\cdot} n_{\cdot j}/n$ [12]. In the latter case, the parameter $n_{\bar{b}j}^e$ is both the mathematical expectation $E(N_{\bar{b}j} | H_0)$ and the variance $\text{var}(N_{\bar{b}j} | H_0)$ of the number of counter-examples under H_0 . It is the number of cases in leaf j that would be counter-examples if they were distributed among the outcome classes with the marginal distribution, i.e. that of the root node (right margin in Table 2).

Gras' *implication index* is the difference $n_{\bar{b}j} - n_{\bar{b}j}^e$ between the observed and expected numbers of counter-examples, standardized by the standard deviation, i.e., if we retain the Poisson model,

$$\text{Imp}(j) = \frac{n_{\bar{b}j} - n_{\bar{b}j}^e}{\sqrt{n_{\bar{b}j}^e}}, \quad (1)$$

which can also be expressed in terms of the number of cases verifying the rules as $\text{Imp}(j) = -(n_{bj} - n_{bj}^e)/\sqrt{n_{\cdot j} - n_{bj}^e}$.

Let us explicit the calculation of the index for our example. We consider for that the variable "predicted class", denoted *cpred*, that takes value 1 for each case (example) belonging to the majority class of its leaf and 0 otherwise (counter-example). By cross-classifying this variable with the premises of the rules, we get Table 3 where the first row gives for each rule its number $n_{\bar{b}j}$ of counter-examples and the second row its number n_{bj} of examples.

Likewise, Table 4 gives the expected numbers $n_{\bar{b}j}^e$ and n_{bj}^e of counter-examples and examples obtained by distributing the $n_{\cdot j}$ covered cases with the marginal distribution. Note that these counts cannot be computed from the margins of Table 3. They are obtained by first dispatching the column total using the marginal distribution of Table 2 and aggregating then separately each resulting column according to its corresponding observed majority class (not the expected one!). This explains why Tables 3 and 4 do not have the same right margin.

Table 3. Observed numbers $n_{\bar{b}j}$ and n_{bj} of counter-examples and examples

Predicted class <i>cpred</i>	Man		Woman		Total
	primary or secondary	tertiary	primary	secondary or tertiary	
0 (counter-example)	23	16	6	28	73
1 (example)	90	12	50	48	200
Total	113	28	56	76	273

Table 4. Expected numbers $n_{\bar{b}j}^e$ and n_{bj}^e of counter-examples and examples

Predicted class <i>cpred</i>	Man		Woman		Total
	primary or secondary	tertiary	primary	secondary or tertiary	
0 (counter-example)	63.33	15.69	31.38	42.59	153
1 (example)	49.67	12.31	24.62	33.41	120
Total	113	28	56	76	273

From these two tables, we get easily the implication indexes using formula (1). They are reported in the first row of Table 5. For the first rule, the index equals $\text{Imp}(1) = -5.068$. This negative value indicates that the number of observed counter-examples is less than the number expected under the independence hypothesis, which stresses the relevance of the rule. For rule 2, the implication index is positive, which tells us that the rule is irrelevant since it generates more counter-examples than classifying without taking account of the condition.

2.3 Implication Index and Residuals

In its formulation (1), the implication index looks like a standardized residual, namely as the (signed root square of) the contribution to the Pearson Chi-square [see for example 1, p 224]. Here, it is indeed the Chi-square that measures the divergence between Tables 3 and 4. These contributions are depicted in Table 5, those of the first row being the implication indexes.

This interpretation of Gras' implication index in terms of residuals (residuals for the fitting of the counts of counter-examples by the independence model) suggests that other forms of residuals used in the framework of the modeling of

Table 5. Contributions to the Chi-square measuring divergence between Tables 3 and 4

Predicted class <i>cpred</i>	Man		Woman	
	primary or secondary	tertiary	primary	secondary or tertiary
0 (counter-example)	-5.068	0.078	-4.531	-2.236
1 (example)	5.722	-0.088	5.116	2.525

Table 6. The various residuals as alternative implication indexes

Residual		Rule R1	Rule R2	Rule R3	Rule R4
standardized (=Imp(j))	res_s	-5.068	0.078	-4.531	-2.236
deviance	res_d	-6.826	0.788	-4.456	-4.847
Freeman-Tukey	res_{FT}	-6.253	0.138	-6.154	-2.414
adjusted	res_a	-9.985	0.124	-7.666	-3.970

the counts in multiway contingency tables could also prove useful for measuring the strength of rules. These include:

The *deviance residual*, $res_d(j) = \text{sign}(n_{\bar{b}j} - n_{\bar{b}j}^e) \sqrt{|2n_{\bar{b}j} \log(n_{\bar{b}j}/n_{\bar{b}j}^e)|}$, which is the signed square root of the contribution (in absolute value) to the likelihood ratio Chi-square [2, pp 136-137]).

Haberman's adjusted residual, $res_a(j) = (n_{\bar{b}j} - n_{\bar{b}j}^e) / \sqrt{n_{\bar{b}j}^e (n_{b\cdot}/n)(1 - n_{\cdot j}/n)}$, which is the Pearson standardized residual divided by its standard error [1, p 224].

Freeman-Tukey's residual, $res_{FT}(j) = \sqrt{n_{\bar{b}j}} + \sqrt{1 + n_{\bar{b}j}} - \sqrt{4n_{\bar{b}j}^e + 1}$, which results from a variance-stabilizing transformation [2, p 137].

Table 6 exhibits the values of theses alternative implication indexes for each of our four rules. We observe that they are concordant as expected. The standardized residual is known to have a less than unity variance. This is because the counts $n_{b\cdot}$ and $n_{\cdot j}$ are sample dependent and hence themselves random. Thus $n_{\bar{b}j}^e$ is only an estimation of the Poisson parameter. Ignoring the randomness of the denominator in formula (1) leads to underestimate the strength. The deviance, adjusted and Freeman-Tukey's residuals are best suited for this situation and are known to have in practice a distribution closer to the standard normal $N(0, 1)$ than the simple standardized residual. We can check in our example that the standardized residuals, i.e. Gras' implication index tends to give lower absolute values than the three alternatives. The only exception is rule R3 which admits only 6 counter-examples.

2.4 Implication Intensity and p -Value

In order to evaluate the statistical significance of the computed implication strength, it is natural to look at the p -value, i.e. at the probability $p(N_{\bar{b}j} \leq n_{\bar{b}j} \mid H_0)$. When $n_{\bar{b}j}^e$ is small, this probability can be obtained, conditionally on $n_{b\cdot}$ and $n_{\cdot j}$, with the Poisson distribution $P(n_{\bar{b}j}^e)$. For large $n_{\bar{b}j}^e$, the normal distribution gives a good approximation. A correction for the continuity may be necessary however, the difference mighting be for example as large as 2.6 points of percentage when $n_{\bar{b}j}^e = 100$. Letting $\phi(\cdot)$ denote the standard normal distribution, we have $p(N_{\bar{b}j} \leq n_{\bar{b}j} \mid H_0) \simeq \phi\left((n_{\bar{b}j} + 0.5 - n_{\bar{b}j}^e) / \sqrt{n_{\bar{b}j}^e}\right)$.

Table 7. The implication intensity and its variants (with continuity correction)

Residual		Rule R1	Rule R2	Rule R3	Rule R4
standardized	res_s	1.000	0.419	1.000	0.985
deviance	res_d	1.000	0.099	1.000	1.000
Freeman-Tukey	res_{FT}	1.000	0.350	1.000	0.988
adjusted	res_a	1.000	0.373	1.000	1.000

The *implication intensity* is defined as the complementary to one of this p -value. Gras [see for instance 7] defines it in terms of the normal approximation, without the correction for continuity however. We compute it as

$$\text{Intens}(j) = 1 - \phi\left((n_{\bar{b}j} + 0.5 - n_{\bar{b}j}^e)/\sqrt{n_{\bar{b}j}^e}\right) . \quad (2)$$

Anyway, this intensity can be interpreted as the probability of getting, under the independence hypothesis H_0 , a higher number of counter-examples than the count observed for rule j . Table 7 gives these intensities for our four rules. It shows also the complementary to 1 of the p -values of the deviance, adjusted and Freeman-Tukey's residuals computed with the continuity correction, i.e. by adding 0.5 to the observed counts of counter-examples.

3 Individual Rule Relevance

The implication intensity and its variants are naturally useful for validating each classification rule individually. This knowledge enriches the usual global validation of the classifier. For example, among the four rules issued from our illustrative tree, rules R1, R3 and R4 are clearly relevant, while R2, with an implication intensity below 50% should clearly be rejected.

The question is then what shall we do with the cases covered by the conditions of irrelevant rules. Two solutions can be envisaged: i) Merging cases covered by an irrelevant rule with another rule, or ii) changing the conclusion. The possible choice of a more suitable conclusion is discussed in Section 4. As for the merging of rules, if we want to respect the tree structure we have indeed to merge cases of a leaf with those of a sister leaf, which is equivalent to pruning the corresponding branch. In our example, this leads to the merge of rules R1 and R2 into a new rule "Man $\Rightarrow$ married". Residuals for the number of counter-examples of this new rules are respectively $res_s = -3.8$, $res_d = -7.1$, $res_{FT} = -4.3$ and $res_a = -8.3$. Except for the deviance residual, they exhibit a slight deterioration as compared to the implicative strength of rule R1.

It is interesting here to compare the implicative quality with the usual error rate used for validating classification rules. The number of counter-examples considered is precisely the number of errors produced by the rule on the learning set. The error rate is thus the percentage of counter-examples among the cases covered by the rule, i.e. $\text{err}(j) = n_{\bar{b}j}/n_{.j}$, which is also equal to $1 - n_{bj}/n_{.j}$, the complementary to one of the confidence. The error rate suffers that from the

Table 8. Implication index penalized for the rule complexity

Rule	res_d	$\ln(n_j)$	k_j	Imp_{pen}
R1	-6.826	4.727	2	-3.75
R2	0.788	3.332	2	3.37
R3	-4.456	4.025	2	-1.62
R4	-4.847	4.331	2	-1.90
Man $\Rightarrow$ married	-7.119	4.949	1	-4.89
Woman $\Rightarrow$ single	-7.271	4.883	1	-5.06

same drawbacks as the confidence. For instance, it does not tell us how better the rule does than a classification independent of any condition. Furthermore, the error rate is linked with the choice of the majority class as conclusion. For our example, the error rate is respectively for our four rules 0.2, 0.57, 0.11 and 0.36. The second rule is thus also the worst from this point of view. Comparing with the error rate at the root node, which is 0.56, shows that this rate of 0.57 is very bad. Thus, for being really informative about the relevance of the rule, the error rate should be compared with the error rate of some naive baseline rule. This is exactly what the implication index does. Resorting to implication indexes, we get in addition probabilities which permits to distinguish between statistically significant and non significant relevance.

Practically, in order to detect over-fitting, error rates are computed on validation data sets or through cross validation. Indeed, the same can be done for the implication quality by computing the implication indexes and intensities in generalization.

Alternatively, we could consider, in the spirit of the BIC (Bayesian information criteria) or MDL (Minimum message length) principle, to penalize the implication index by the complexity of the condition. Since the lower the implication index of a rule j , the best it is, the index should be penalized by the length k_j of the branch that defines the condition of rule j . The general idea behind such penalization is that the simpler the condition, the lower the risk to assign a bad distribution to a case. As a first proposal we suggest the following penalized form inspired from the BIC [13] and based on the deviance residual

$$Imp_{pen}(j) = res_d(j) + \sqrt{k_j \ln(n_j)} .$$

For our example, the values of the penalized index are given in Table 8.

This penalized form of the index confirms the ranking of the initial rules, which here have all the same length $k_j = 2$. In addition, it is useful for validating the merge of the two rules R1 and R2. Table 8 highlights the superiority of the merged rule “Man $\Rightarrow$ married” over both rules R1 and R2. It gives a clear signal in favor of the merge.

At the root node, both the residual and the number of conditions are zero. Hence, the penalized implication index is zero too. Thus, a positive penalized implication index suggests that we can hardly expect that the rule would do better

Table 9. Implication indexes and intensities of rule 2 for each possible conclusion

Residual		Indexes			Intensity		
		married	single	div./wid.	married	single	div./wid.
Standardized	res_s	1.6	0.1	-1.3	0.043	0.419	0.891
Deviance	res_d	3.9	0.8	-3.4	0.000	0.099	0.999
Freeman-Tukey	res_{FT}	1.5	0.1	-1.4	0.054	0.398	0.895
Adjusted	res_a	2.4	0.1	-2.0	0.005	0.379	0.968

in generalization than a random classification independent of any condition. For our example, this confirms once again the badness of rule R2.

4 Implication Strength Versus Majority Rule

A final aspect regarding the implication strength is its link with the choice of the conclusion for each rule. The idea, that has for instance been exploited in [15, pp 282-287], is to chose in each leaf the conclusion for which the rule gets its highest implication intensity. This is indeed an alternative to the majority rule.

To illustrate, we give in Table 9 the values of the alternative indexes and intensities of implication for each of the three possible conclusion that may be assigned to rule 2. The conclusion labeled “single” corresponds to the majority class. We see here that there is a better conclusion from the strength of implication standpoint, namely “divorced or widowed”. All four indexes designate this conclusion as the best with an implication intensity that goes from 89.1% for Gras’ index to 99.9% for the deviance residual. Again we can notice that Gras’ index seems to slightly under-estimate the implication intensity.

5 Conclusion

The aim of the paper was to demonstrate the usefulness of the concept of implication strength for classification rules derived from induced decision trees. We have shown that it may prove helpful for evaluating the individual relevance of rules as well as an alternative to the majority class rule for selecting the rule conclusion. We have also stressed the interest of considering implication indexes penalized for the complexity.

Much remains to be done. The variants to Gras’ index, the continuity corrections, the penalized index all would merit further theoretical as well as empirical study of their impact in real word problems. In Section 4, we have stressed the interest of the implication strength approach for selecting the most appropriate conclusion for each leaf. We should even be able to go further and use criteria based on implication indexes in the tree growing process. An other important issue that we plan to investigate is the relationship between the individual performance of the rule and the global classifier efficiency.

References

- [1] Agresti, A. *Categorical Data Analysis*. Wiley, New York, 1990.
- [2] Bishop, Y.M.M., Fienberg, S.E., Holland, P.W. *Discrete Multivariate Analysis*. MIT Press, Cambridge MA, 1975.
- [3] Blanchard, J., Guillet, F., Gras, R., Briand, H. Using information-theoretic measures to assess association rule interestingness. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005)*, pages 66–73. IEEE Computer Society, 2005. ISBN 0-7695-2278-5.
- [4] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J. *Classification And Regression Trees*. Chapman and Hall, New York, 1984.
- [5] Briand, H., Fleury, L., Gras, R., Masson, Y., Philippe, J. A statistical measure of rules strength for machine learning. In *Proceedings of the Second World Conference on the Fundamentals of Artificial Intelligence (WOCFAI 1995)*, pages 51–62, Paris, 1995. Angkor.
- [6] Gras, R. *Contribution à l'étude expérimentale et à l'analyse de certaines acquisitions cognitives et de certains objectifs didactiques*. Thèse d'état, Université de Rennes 1, France, 1979.
- [7] Gras, R., Couturier, R., Blanchard, J., Briand, H., Kuntz, P., Peter, P. Quelques critères pour une mesure de qualité de règles d'association. *Revue des nouvelles technologies de l'information RNTI*, E-1:3–30, 2004.
- [8] Gras, R., Larher, A. L'implication statistique, une nouvelle méthode d'analyse de données. *Mathématique, Informatique et Sciences Humaines*, (120):5–31, 1992.
- [9] Gras, R., Ratsima-Rajohn, H. L'implication statistique, une nouvelle méthode d'analyse de données. *RAIRO Recherche Opérationnelle*, 30(3):217–232, 1996.
- [10] Guillaume, S., Guillet, F., Philippé, J. Improving the discovery of association rules with intensity of implication. In Zytzkow, J.M., Quafafou, M., editors, *Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 1998)*, volume 1510 of *Lecture Notes in Computer Science*, pages 318–327. Springer, 1998. ISBN 3-540-65068-7.
- [11] Kass, G.V. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2):119–127, 1980.
- [12] Lerman, I.C., Gras, R., Rostam, H. Elaboration d'un indice d'implication pour données binaires I. *Mathématiques et sciences humaines*, (74):5–35, 1981.
- [13] Raftery, A.E. Bayesian model selection in social research. In Marsden, P., editor, *Sociological Methodology*, pages 111–163. The American Sociological Association, Washington, DC, 1995.
- [14] Suzuki, E., Kodratoff, Y. Discovery of surprising exception rules based on intensity of implication. In Zytzkow, J.M., Quafafou, M., editors, *Principles of Data Mining and Knowledge Discovery, Second European Symposium, PKDD '98, Nantes, France, September 23-26, Proceedings*, pages 10–18. Springer, Berlin, 1998.
- [15] Zighed, D.A., Rakotomalala, R. *Graphes d'induction: apprentissage et data mining*. Hermes Science Publications, Paris, 2000.

A New Clustering Approach for Symbolic Data and Its Validation: Application to the Healthcare Data

Haytham Elghazel¹, Véronique Deslandres¹, Mohand-Said Hacid²,
Alain Dussauchoy¹, and Hamamache Kheddouci¹

¹ PRISMa Laboratory, Claude Bernard University of Lyon I, 43 Bd du 11 novembre 1918,
69622 Villeurbanne cedex, France

{elghazel, deslandres, dussauchoy, hkheddou}@univ-lyon1.fr

² LIRIS Laboratory, Claude Bernard University of Lyon I, 43 Bd du 11 novembre 1918,
69622 Villeurbanne cedex, France
mshacid@liris.cnrs.fr

Abstract. Graph coloring is used to characterize some properties of graphs. A *b-coloring* of a graph G (using colors $1, 2, \dots, k$) is a coloring of the vertices of G such that (i) two neighbors have different colors (proper coloring) and (ii) for each color class there exists a dominating vertex which is adjacent to all other $k-1$ color classes. In this paper, based on a *b-coloring* of a graph, we propose a new clustering technique. Additionally, we provide a cluster validation algorithm. This algorithm aims at finding the optimal number of clusters by evaluating the property of *color dominating vertex*. We adopt this clustering technique for discovering a new typology of hospital stays in the French healthcare system.

1 Introduction

Clustering is the process of dividing a set of given objects into groups, or clusters, such that all objects in the same group are similar to each other, while objects from different groups are dissimilar. Clustering plays an important role in data mining applications such as Web analysis, information retrieval and many other domains.

In French hospitals, the Diagnosis Related Groups (DRG) system was introduced in the eighties, by the Information Systems Medicalisation Program (PMSI). According to the PMSI process deployment, the PMSI data can also be used for the assessment of performances for both public and private hospitals. This evaluation is based on the production of a Standard Discharge Summary (RSS) for each hospital stay. The RSS contains data related to the nature of treatments, medical exams and diagnosis as well as patient's information. The RSS is then identified to correspond to one patient's group called DRG, which is used for the classification of hospital stays. This operation is performed using a supervised approach according to a decision tree.

In spite of the successive improvements of DRG classification, this method remains inefficient for public and private hospitals. The major known problem of this method is that heterogeneous pathologies and examinations find themselves in the same DRG class. In this paper, we propose a new clustering approach based on graph *b-coloring* as an alternative for DRG classification: it builds a refined typology of

hospital stays. A graph *b-coloring* consists to color the vertices of a graph with the largest number of color such that (i) two neighbors have different colors (proper coloring) and (ii) for each color class there exists a *dominating vertex* which is adjacent to all other color classes.

The paper is organized as follows: Section 2 provides summary of related work. Section 3 describes the clustering algorithm that uses as foundation a *graph b-coloring* method. Section 4 is devoted to the cluster validation algorithm. In Section 5, the application of the proposed method on *PMSI* data is illustrated. The experiments show that the algorithm is appropriate to find the optimal number of *hospital stay* groups. We conclude in Section 6 by anticipating on necessary extensions.

2 Related Work

Clustering of data is generally based on two approaches: *hierarchical and partitioning* [1]. The *hierarchical* clustering algorithms build a cluster hierarchy or, in other words, a tree of clusters, (*dendrogram*) whose leaves are the data points and whose internal nodes represent nested clusters of various sizes [2]. Hierarchical clustering methods can be further subdivided into *agglomerative* and *divisive*, depending on whether they start from the bottom or the top of the tree. The main weakness of this algorithm is a choice of the level that provides the best partition. It is often the data miner who decides about the number of clusters associated both the context and the goal of the analysis. Among these hierarchical methods, some authors have proposed to use *graph coloring techniques* for the classification purpose. In [3], the authors propose a divisive classification method based on the distance tables, where the iterative algorithm consists, at each step, in finding a partition by subdivising the class with the largest diameter into two classes in order to offer a new partition with minimal diameter. The subdivision is obtained by a *2-coloring* of the vertices of the maximum spanning tree built from the dissimilarity table.

The *partitioning* clustering algorithms give a single partition of the data by fixing some parameters (number of clusters, thresholds, etc.). The partitioning algorithm typically starts with an initial partition of data set and then uses an iterative control strategy to optimize an objective function. Each cluster is then represented by its centroid (*k-means algorithms*: does not work well with symbolic data like medical diagnoses) [4] or by one of its objects located near its center (*k-medoid algorithms*) [5]. In practice, the algorithm is typically running multiple times with different starting states to identify the optimal values of the fixed parameters. The best configuration obtained from all of the runs is used as the output clustering. In the framework of partitioning methods, Hansen and Delattre [6] reduced the partitioning problem of a data set into p classes with minimal diameter, to the *minimal coloring* problem of a *superior threshold graph*. The edges of this graph are the pairs of vertices distanced from more than a given threshold.

In this paper we propose a new coloring method namely *b-coloring*, well-adapted for clustering. It allows to build a fine partition of the data set (*classical* or *symbolic*) in classes where the number of classes is not pre-defined. This approach is interesting

in the sense that it identifies each cluster by at least one *dominant object* which guarantees the disparity between the classes of the partition. The optimal number of classes is then determined based on a new validation algorithm. This algorithm is designed to identify a partition of data that are compact and well separated by evaluating the property of *class dominant object*.

3 Symbolic Clustering Approach

In this section we describe our new graph clustering algorithm. It uses pairwise representation, where the objects (*hospital stays*) are mapped to the nodes of an undirected edge-weighted graph. The edge weights reflect the dissimilarity between the corresponding pair of nodes. The objective here is to generate a partition of a set of n objects $V = \{v_1, \dots, v_n\}$ in cohesive and well separated classes where their number is not given beforehand. The set V is described by a dissimilarity table $D = \{d_{ij} \mid v_i, v_j \in V\}$.

Notations: we denote by $G=(V,D)$, the complete weighted graph depicting the objects set $V=\{v_1, \dots, v_n\}$ where each pair (v_i, v_j) is weighted by the dissimilarity d_{ij} . Each object v_i is represented by a mixture of m symbolic as well as classical attributes denoted x_i^k ($k \in \{1 \dots m\}$). Therefore, the object v_i corresponds to the vector $(x_i^1, x_i^2, \dots, x_i^m)$. *Classical attributes* are defined by either quantitative or qualitative values when *symbolic attributes* can be either a set of values or intervals. The clustering problem is now formulated as a graph *b-coloring* problem.

3.1 A Graph b-Coloring Method

Let $G=(V,E)$ be an undirected, connected and simple graph without loops with a vertex set V and an edge set E . The *b-coloring* of G is the vertex coloring such that:

- For each pair of adjacent vertices $(x,y) \in G$, the color of x and y are different (*proper coloring*),
- In each color class, there exists at least one vertex having neighbors in all other color classes. Such a vertex is called a *dominating vertex*. A color which has a dominating vertex is called a *dominating color*.

We call the *b-chromatic number* $\phi(G)$ of a graph G the largest number k such that G has a *b-coloring* with k colors.

In contrast to the *minimal coloring* of the vertices of graph G , the *b-coloring* consists to a *maximal coloring* under property and dominance constraints.

The degree of vertex is the number of its neighbors and the maximum degree denoted by $\Delta(G)$ of a graph G is the largest degree over all vertices of G . Irving and Manlove [7] have shown that:

$$\chi(G) \leq \phi(G) \leq \Delta(G) + 1. \quad (1)$$

where *chromatic number* $\chi(G)$ is the minimum number of colors used to have a proper coloring of G ,

Irving and Manlove have also shown that finding the *b-chromatic number* $\phi(G)$ for any graph is a *NP-hard* problem. Several authors investigated this parameter for particular classes of graphs like trees [7] and power graphs [8]

3.2 Construction of the Threshold Graph

Our clustering approach based on a *b-coloring* technique requires to construct a *superior threshold graph*, which is a partial graph of the initial graph $G=(V,D)$. Let $G_{>s}=(V,D_{>s})$ be the superior threshold graph associated with threshold value s chosen among the dissimilarity table D . In other words, $G_{>s}$ is given by $V=\{v_1, \dots, v_n\}$ as vertex set and $\{(v_i, v_j) | D(v_i, v_j)=d_{ij} > s\}$ as edge set. Figure 1 gives the superior threshold graph $G_{>0.2}$ associated with the following dissimilarity table:

Table 1. One dissimilarity table

v_i	A	B	C	D	E	F	G
A	0						
B	0.20	0					
C	0.20	0.30	0				
D	0.10	0.20	0.25	0			
E	0.20	0.20	0.10	0.40	0		
F	0.20	0.20	0.20	0.25	0.20	0	
G	0.25	0.20	0.20	0.10	0.20	0.65	0

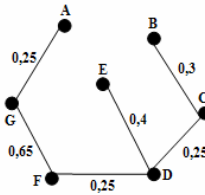


Fig. 1. The superior threshold graph $G_{>0.2}$

3.3 Algorithm

We present here our new clustering algorithm which is based on a graph *b-coloring* technique. The idea consists in applying the *b-coloring technique*-which we adapt to the clustering problem, on the superior threshold graph $G_{>s}$. One partition associated to the selected threshold value s is then returned. Doing so, the dissimilarity between objects within the same class is lower to s .

Notations: Let $G=(V,D_{>s})$ be the threshold graph, such that:

- $V=\{v_1, \dots, v_n\}$: the vertex set and $D_{>s}$ the edge set.
- Δ : the maximum degree of G .
- $c(v_i)$: the color (integer value) of the vertex v_i .
- $N(v_i)$: the neighborhood of vertex v_i .
- $N_c(v_i)$: the neighborhood colors of vertex v_i .
- $dist(v_i, c)$: the distance between the vertex v_i and the color c defined as the minimum distance from v_i to all vertices with color c :

$$dist(v_i, c) = \min \left\{ \begin{array}{l} d_{i,j} = D(v_i, v_j) \\ 1 \leq i \neq j \leq n \text{ and } c(v_j) = c \end{array} \right\} \quad (2)$$

- L is the color set used in the graph (a set of integer values).
- D_m is a set of colors which have dominating vertices.
- ND_m is a set of colors which have not dominating vertices.

A Graph b-Coloring Algorithm

The *b-coloring* algorithm uses a two-step procedure. Before starting the *b-coloring* algorithm, each vertex must be colored. The following procedure gives an initial configuration using the maximum number of colors available for a *b-coloring* (Eq. 2) (i.e. $\Delta+1$). While vertices are not colored yet, the algorithm starts from the vertex of maximum degree Δ (let v be such a vertex). Then the algorithm puts: $c(v)=1$. The *Procedure 1* then colors remaining vertices. Let:

- T be the set of colored vertices which is sorted on descending order of vertex degrees. Initially T contains only the vertex v and it is updated as long as *Procedure 1* runs.
- $Update(N_c(v_i))$ be the method which updates the neighborhood colors of the vertex v_i when the color of at least one of its neighbors is changed.
- $Add(o, S)$ be the method which adds o (e.g.: color, vertex) into the set S .
- $Remove(o, S)$ be the method which remove the object o from the set S .
- $Sort(S)$ be the method which sorts the elements of S by decreasing order of vertex degree.

Procedure 1 Init_b-coloring()

begin

$L := \{1, 2, \dots, \Delta+1\};$

repeat

select v_i from T ; $M := N_c(v_i) \cup \{c(v_i)\}$; $q := 0$;

for each vertex $v_j \in N(v_i)$ such that $c(v_j) = \emptyset$ do

$q := \min\{k \mid k > q, k \notin M \text{ and } k \notin N_c(v_j)\};$

if $q \leq \Delta+1$ then $c(v_j) := q$;

else $c(v_j) := \min\{k \mid k \notin N_c(v_j)\};$

$Add(v_j, T)$;

for each vertex $v_k \in N(v_j)$ do $Update(N_c(v_k))$; enddo.

$sort(T)$;

enddo.

if $N_c(v_i) = L \setminus \{c(v_i)\}$ then $Add(c(v_i), D_m)$; endif.

$Remove(v_i, T)$;

until $(T = \emptyset)$

end.

Lemma 1. *Procedure 1* executes in $O(n^2 \Delta)$.

Proof. *Procedure 1* is applied once on every colored vertex (at most n). First, the algorithm colors every non-colored neighbor (at most Δ). For each neighbor, the change of color is propagated to its own neighbors and the elements of set T (at most n : $O(n)$) are sorted by decreasing order of degree. Therefore *Procedure 1* executes in at most $O(n^2 \Delta)$.

Procedure 1 performed on the previous threshold graph $G_{>0.2}$ gives the following colored graph:

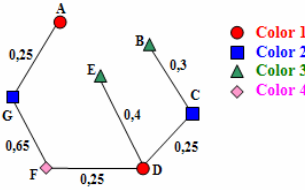


Fig. 2. The initial colored graph $G_{>0.2}$

After applying *Procedure 1*, some colors remain without any *dominating vertex*. The following *Procedure 2* finds a *b-coloring* of graph G where all the colors belonging to L are *dominating colors* (i.e. $D_m=L$). The idea is the following: each non-dominating color q (i.e. $q \in ND_m$) can be changed. In fact, after removing q from the graph G , for each vertex v_i colored with q (i.e. $c(v_i)=q$), a new color is assigned to v_i which is different from those of its neighborhoods. As our objective is to find a partition such that the sum of vertex dissimilarities within each class is minimized, the color whose distance with v_i is minimal will be selected if there is a choice between many colors for v_i . Before starting again with another color $q' \in ND_m$, we verify if colors of ND_m have a dominating vertex (in such a case, these colors are added to the D_m set).

Procedure 2 find_b-coloring()

```

begin
  repeat
     $q := \max\{k \mid k \in ND_m\}$ ;  $L := L \setminus \{q\}$ ;  $ND_m := L \setminus D_m$ ;
    for each vertex  $v_i$  such that  $c(v_i)=q$  do
       $K := \{k \mid k \in L \text{ and } k \notin N_c(v_i)\}$ ;
       $c(v_i) := \{c \mid \text{dist}(v_i, c) = \min_{k \in K} (\text{dist}(v_i, k))\}$ ;
    enddo.
    I { for each vertex  $v_j$  such that  $c(v_j) \in ND_m$  do
        Update( $N_c(v_j)$ );
        if  $N_c(v_j) = L \setminus \{c(v_j)\}$  then Add( $c(v_j), D_m$ ); endif.
      } enddo.
    until ( $ND_m = \emptyset$ )
  end.

```

Once we modify the color of all vertices v_i having color q , we must verify, using the instructions I, if each non-dominating color became a new dominating color.

Lemma 2. We see easily that the coloring generated by *Procedure 2* is *proper*.

Proof. If the color $c(v_i)$ is changed, it is selected to be different from the neighbor colors of v_i .

Lemma 3. After running the *Procedure 2*, there exists at least one dominating vertex for each obtained color class.

Proof. The *Procedure 2* converges when the set of colors which have not dominating vertices is an empty set.

Lemma 4. *Procedure 2* generates the *b-coloring* of any graph G in $O(n\Delta^2)$.

Proof. *Procedure 2* is applied for every color q without dominating vertices (at most Δ). A color q is removed, and every vertex v_i such that $c(v_i)=q$ (at most n) is recolored. Then, for every vertex v_j (at most n), we update its neighborhood colors $N_c(v_j)$ (at most Δ) and thus we verify if its color is a new dominating color. Therefore *Procedure 2* uses at most $((n+n*\Delta) * \Delta)$ instructions. Its complexity is $O(n\Delta^2)$.

Let us consider the previous example. Starting with $L=\{1,2,3,4\}$ and $D_m=\{1\}$, the b -coloring of the graph $G_{>0.2}$ given by *Procedure 2* is depicted in Figure 3.

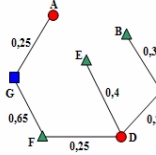


Fig. 3. The b -coloring of the graph $G_{>0.2}$

Thus, the graph $G_{>0.2}$ is colored with three colors, with one *dominating vertex* for each color. Therefore, the partition returned by the algorithm associated to the threshold 0.2 is composed of the three following classes: $C_1=\{\mathbf{D},\mathbf{A}\}$, $C_2=\{\mathbf{C},\mathbf{G}\}$ and $C_3=\{\mathbf{F},\mathbf{B},\mathbf{E}\}$ as shown in Figure 4. The bold characters represent dominating vertices of the obtained classes. This means that these vertices are joined to at least one vertex in each other color class.

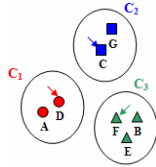


Fig. 4. A partition associated to the threshold 0.2

4 Cluster Validation Algorithm

The clustering algorithm is an iterative algorithm which that executes multiple runs increasing the value of dissimilarity threshold s . Next, the major problem is the validation of clusters resulting from all the runs to select the best configuration (partition) which is considered as the optimal output clustering.

Different cluster validation indices have been proposed in the literature [9], such as *Davies-Bouldin index* and *Dunn's index*. Therefore, the correct value of dissimilarity threshold corresponds to an optimal index value. However, there are several ties or near-ties corresponding to the best results. In order to make the automatic determination of the optimal data partition more robust and eliminate such ties, a cluster validation algorithm was implemented by evaluating the property of dominance in each cluster from all the best considered threshold partitions.

We denote by $I_V(P_i)$ the validity index value associated with the partition P_i .

Algorithm

```

begin
V=(v1, v2, ..., vn);
for each Partition Pi selected as one best partition do
  for each cluster Ci=(v1i, ..., vηii)∈ Pi where v1i is a dominant do
    for each object vji∈ Ci do
      Q(v1i -> vji) := Q(v1i -> vji) + Iv(Pi); // The Rule quality
      Add(v1i -> vji) to the rules set R if it does not exists;
    enddo. enddo. enddo.
Sort R by decreasing order of rule quality Q;
for each rule (x -> y) ∈ R taken consecutively such that v ≠ ∅ do
  Add y to the cluster C which contains x;
  Remove x and y from V if they exist;
  Remove from R all the rules such that x or y are results;
  For each rule Ri such that y is dominant as (y -> t) do
    Q(y -> t) := Q(x -> t) / Q(y -> t);
  enddo.
  Sort R by decreasing order of rule quality Q;
enddo.
end.

```

The cluster validation algorithm was run on the previous data set (A,B,C,D,E,F,G). For each threshold selected in the dissimilarity Table 1, the value of *Dunn's index* of the returned partition is calculated (cf. Table 2). The Dunn's index is designed to offer a compromise between the *intercluster separation* and the *intracluster distances*. It identifies sets of clusters that are compact and well separated. Let $s_a(C_k)$ denote the *average distance* within the cluster C_k and $d_a(C_k, C_l)$ the *between-cluster separation*. The larger values of Dunn's indicate the better clustering. For a partition P into p clusters $(C_1, C_2, \dots, C_p)$, the Dunn's index is defined as follows:

$$Dunn = \min_i \left\{ \min_{j \neq i} \left\{ \frac{d_a(C_i, C_j)}{\max_k s_a(C_k)} \right\} \right\} \quad \text{where} \quad \begin{cases} d_a(C_i, C_j) = \frac{1}{\eta_i \times \eta_j} \sum_{u=1}^{\eta_i} \sum_{q=1}^{\eta_j} D(v_u, v_q) \\ s_a(C_k) = \frac{1}{\eta_k \times (\eta_k - 1)} \sum_{u=1}^{\eta_k} \sum_{q=1}^{\eta_k} D(v_u, v_q) \end{cases} \quad (3)$$

where $1 \leq i, j, k \leq p$, D indicates the dissimilarity measures, and η_k the number of objects in cluster C_k .

Table 2. Evaluation of each threshold partition

Threshold	Partition	Dunn's index value
0.1	{A} {B} {C,E}{G,D}{F}	1.75
0.2	{C,G} {F,B,E} {D,A}	1.00
0.25	{F,A,B,D}{C,E,G}	1.37
0.3	{F,A,B,C,D}{E,G}	1.19
0.4	{F,A,B,C,D,E} {G}	1.25

From Table 2, one can note that the optimal partition which maximizes the Dunn's index value is associated with the threshold 0.1. However, if there are several ties or near-ties corresponding to the maximal values of Dunn's index, we propose to apply

the validation algorithm on the first closest partitions. Assume that the first four best results for the Dunn's index (1.75, 1.37, 1.25 and 1.19) are too closed, so it is not easy to really decide which is the optimal partition, the cluster validation algorithm gives the partition composed of the two clusters: {F,A,B,D} and {C,E,G}.

5 Experiments

The clustering algorithm and the cluster validation algorithm described in Sections 3 and 4, respectively, were implemented and evaluated. The clustering algorithm was applied to a real data set. This data set consists of 500 instances of *PMSI hospital stays* with 10 features. There are 5 quantitative, 2 qualitative and 3 symbolic attributes. The instances are pre-classified on 21 DRG.

Since the objective was to perform unsupervised classification, hence any class information was eliminated from the data. The results are compared with that of Agglomerative Hierarchical Classification (AHC) [2] and the predefined DRG classification. The measures of the overall average distance within cluster S_a and the overall average between-cluster separations D_a are used for this purpose. For a partition P into p clusters ($C_1, C_2, \dots, C_p$) the values of S_a and D_a are given by:

$$S_a = \frac{1}{p} \sum_{i=1}^p s_a(C_i) \quad \text{and} \quad D_a = \frac{1}{p \times (p-1)} \sum_{i=1}^p \sum_{j=1}^p d_a(C_i, C_j) \quad (4)$$

Table 3 shows the first ten best clustering results according to the larger values of Dunn's index.

Table 3. The first ten best Dunn's index values of the PMSI data

Threshold	Number of clusters of partition	Dunn's index value
1.540665	28	0.568483
1.360186	45	0.568007
1.539183	27	0.567937
1.469705	32	0.567813
1.363826	45	0.566491
1.581157	27	0.562751
1.583693	26	0.562511
1.590260	25	0.561748
1.591159	24	0.561563
1.377407	43	0.560905

Table 4. Evaluation of AHC, DRG and our typology of PMSI hospital stays

Classification Method	Number of clusters	Dunn's	S_a	D_a	D_a/S_a
Our optimal partition	27	0.5679	1.1767	1.8729	1.5916
Optimal AHC partition	23	0.4026	1.2490	1.8516	1.4824
DRG	21	0.2843	1.2932	1.7989	1.3910

The application of the cluster validation algorithm to these ten partitions provides the partition composed of 27 clusters and associated with the threshold 1.539183 as an optimal partition. According to the *Dunn's index*, S_a and D_a measures, the results presented in the table 4 show that our clustering approach gives better results than the *AHC* method and also the *DRG* classification.

6 Conclusion

In this paper, a new clustering algorithm is proposed. It is based on a graph *b-coloring* technique. We implemented, performed experiments, and compared our approach to the Agglomerative Hierarchical Classification and the DRG classification, and illustrated its efficiency on the *PMSI* data. A real advantage of this method is that it offers a real representation of clusters by a dominant object which is used to evaluate the quality of cluster. So, an optimization algorithm has been developed based on a cluster dominance property. The optimal partition is therefore automatically identified.

The present work can be extended in many directions: (1) we are leading additional experiments on a larger *PMSI* data set by considering other validity indices, (2) improvements have to be made on this stage in order to formally prove the performance of the method; (3) the validation stage of the *hospital stays typology* will be completed with the participation of several specialists from the medical domain.

References

- [1] Jain, A.K., M.N. Murty, and P.J. Flynn. Data Clustering: A Review. In *ACM Computing Surveys*, volume. 31, pages 264-323, 1999.
- [2] Guha, S., R. Rastogi, and K. Shim. CURE: An efficient clustering algorithm for large databases. In *Proceedings of the ACM SIGMOD Conference*, Seattle, WA, pages 73-84, 1998.
- [3] Guénoche, A., P. Hansen, and B. Jaumard. Efficient algorithms for divisive hierarchical clustering with the diameter criterion. *Journal of Classification*, volume. 8, pages 5-30, 1991.
- [4] Hartigan, J. and M. Wong. Algorithm AS136: A k-means clustering algorithm. *Journal of Applied Statistics*, volume 28, pages 100-108, 1979.
- [5] NG, R. and J. HAN. Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th Conference on VLDB*, Santiago, Chile, pages 144-155, 1994.
- [6] Hansen, P. and M. Delattre. Complete-link cluster Analysis by graph coloring. *Journal of the American Statistical Association*, volume 73, pages 397-403, 1978.
- [7] Irving, W. and D. F. Manlove. The b-chromatic number of a graph. *Discrete Applied Mathematics*, volume 91, pages 127-141, 1999.
- [8] Effantin, B. and H. Kheddouci. The b-chromatic number of some power graphs. *Discrete Mathematics and Theoretical Computer Science*, 6(1):45-54, 2003.
- [9] Bezdek, J.C. and N.R. Pal. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3):301-315, 1998.

Action Rules Discovery System DEAR_3

Li-Shiang Tsay¹ and Zbigniew W. Ras^{2,3}

¹ Hampton Univ., Computer Science Dept., Hampton VA 23668, USA

² Univ. of North Carolina, Computer Science Dept., Charlotte, NC 28223, USA

³ Polish Academy of Sciences, ICS, Ordona 21, 01-237 Warsaw, Poland

Abstract. E-action rules, introduced in [8], represent actionability knowledge hidden in a decision system. They enhance action rules [3] and extended action rules [4], [6], [7] by assuming that data can be either symbolic or nominal. Several efficient strategies for mining e-action rules have been developed [6], [7], [5], and [8]. All of them assume that data are complete. Clearly, this constraint has to be relaxed since information about attribute values for some objects can be missing or represented as multi-values. To solve this problem, we present DEAR_3 which is an e-action rule generating algorithm. It has three major improvements in comparison to DEAR_2: handling data with missing attribute values and uncertain attribute values, and pruning outliers at its earlier stage.

1 Introduction

E-action rule mining [8], [4], [5] is a strategy for discovering, from a database, certain type of expressions useful for solving actionability issues. It evaluates classification rules to build a strategy of action based on condition features in order to get a desired effect on a decision feature. An actionable strategy is represented as $[(\omega) \wedge (\alpha \rightarrow \beta)] \Rightarrow [\phi \rightarrow \psi]$, where ω , $(\alpha \rightarrow \beta)$, and $[\phi \rightarrow \psi]$ are events. It states that when the fixed condition ω is satisfied and the changeable behavior $(\alpha \rightarrow \beta)$ occurs in tuples from a database so does the expectation $[\phi \rightarrow \psi]$. Support and confidence are used to measure the importance of each strategy to avoid generating irrelevant, spurious, and insignificant rules. The representation of an e-action rule is easy to understand and can be applied in many real-life applications as a powerful solution to reclassification problems.

The goal of e-action rules is to augment people's ability to deliver solutions which can maximize their data assets. Its concept was introduced in [8] to enhance action rules [3] and extended action rules [4], [6], and [7] used to represent actionable knowledge hidden in a database. Several efficient algorithms for mining e-action rules have been developed [6], [7], [5], and [8]. All of them assume that information about objects in a database is complete. Clearly, this constraint has to be relaxed since attribute values for some objects can be missing or represented as multi-values. To approach the problem of incomplete values, two options can be considered: [1] discard all samples in a dataset with imperfect data, [2] define a new algorithm or modify an existing algorithm that will work with incomplete values. The first option is simple but unacceptable especially

when large amount of incomplete values exist in a dataset. DEAR_3, presented in this paper, is an e-action rule generating algorithm. It has three major improvements in comparison to DEAR_2: handling data with missing attribute values, handling data with uncertain attribute values, and pruning outliers at its earlier stage. We start with recalling some fundamental concepts such as an incomplete information system and an incomplete decision table.

2 Incomplete Information Systems and E-Action Rules

By an incomplete information system (IS), introduced in [1], we mean $IS = (U, A, V)$, where:

- U is a finite set of objects,
- A is a finite set of attributes, and
- $V = \bigcup \{V_a : a \in A\}$ is a finite set of attribute values. V_a is a domain of attribute a . Additionally, we assume that for each attribute $a \in A$ and $x \in U$, $a(x) = \{(a_i, p_i) : i \in J_{a(x)} \wedge (\forall i \in J_{a(x)})[a_i \in V_a] \wedge \sum p_i = 1\}$. By p_i we mean the confidence in a value a_i for x .

To give an example, objects can be interpreted as customers, attributes as features such as, *offers made by a bank*, *nationality*, *age*, *number of children*, etc. The set $\{(mexican, 60\%), (spanish, 40\%)\}$ can be seen as a value of the attribute *nationality*. If the incompleteness means *null value*, DEAR_3 can generate a set of promising attribute values with varied weights as a replacement for the missing value.

We only consider a special type of information systems, called decision tables [2]. Their sets of attributes are partitioned into conditions and decisions. Additionally, we assume that the set of conditions is partitioned into stable conditions and flexible conditions. For simplicity reason, we assume that there is only one decision attribute. *Racial* is an example of a stable attribute. *Interest Rate* on any customer account is an example of a flexible attribute as the bank can adjust rates. We adopt the following definition of a decision table:

By a decision table we mean an information system $IS = \{U, A_{St} \cup A_{Fl} \cup \{d\}, V\}$, where:

- U is a set of objects,
- $A_{St} = \{a_{St}[i] : 1 \leq i \leq I\}$ is a set of stable attributes, and $V_i^{St} = \{a_{St}[i, j] : 1 \leq j \leq J(i)\}$ is the domain of attribute $a_{St}[i]$.
- $A_{Fl} = \{a_{Fl}[i] : 1 \leq i \leq L\}$ is a set of flexible attributes, and $V_i^{Fl} = \{a_{Fl}[i, j] : 1 \leq j \leq L(i)\}$ is the domain of attribute $a_{Fl}[i]$.
- d is a decision attribute where $V_d = \{d[m] : 1 \leq m \leq M\}$.

For simplicity reason, the term $a(i_1, j_1) \cdot a(i_2, j_2) \cdot \dots \cdot a(i_r, j_r)$ is denoted by $[a(i_k, j_k)]_{k \in \{1, 2, \dots, r\}}$, where all $i_1, i_2, \dots, i_r$ are distinct values and $j_p \leq J(i_p), 1 \leq p \leq r$. As an example of an incomplete decision table we take $IS = (\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}, \{c, b\} \cup \{e\} \cup \{d\})$ represented by Table 1. The

Table 1. An Incomplete Decision System (IS)

U	e	b	c	d
x_1	$\{(4, 0.33), (2, 0.67)\}$	1		L
x_2	$\{(4, 0.25), (1, 0.75)\}$	$\{(1, 0.5), (5, 0.5)\}$	$\{(1, 0.33), (4, 0.67)\}$	L
x_3		$\{(1, 0.5), (2, 0.5)\}$	1	L
x_4	1	3	$\{(1, 0.5), (3, 0.5)\}$	H
x_5	$\{(3, 0.67), (5, 0.33)\}$		1	H
x_6	2	$\{(1, 0.9), (6, 0.1)\}$	2	L
x_7		$\{(1, 0.33), (4, 0.67)\}$	$\{(1, 0.25), (2, 0.75)\}$	H
x_8	1	3	2	H
x_9	$\{(2, 0.8), (1, 0.2)\}$	1		L

set $\{c, b\}$ lists flexible attributes, e is a stable attribute and d is a decision attribute.

By $L(r)$ we mean all attributes listed in the IF part of a rule r . For example, if $r_1 = [(a_1, 2) \wedge (a_2, 1) \wedge (a_3, 4) \rightarrow (d, 8)]$ is a rule then $L(r_1) = \{a_1, a_2, a_3\}$.

By $d(r_1)$ we denote the decision value of that rule. In our example $d(r_1) = 8$. If r_1, r_2 are rules and $B \subseteq A_{St} \cup A_{Fl}$ is a set of attributes, then $r_1/B = r_2/B$ means that the conditional parts of rules r_1, r_2 restricted to attributes B are the same. For example if $r_2 = [(a_2, 1) * (a_3, 4) \rightarrow (d, 1)]$, then $r_1/\{a_2, a_3\} = r_2/\{a_2, a_3\}$.

Now, let us assume that $(a, v \rightarrow w)$ denotes the fact that the value of attribute a has been changed from v to w . Similarly, the term $(a, v \rightarrow w)(x)$ means that $a(x) = v$ has been changed to $a(x) = w$. That is, the property (a, v) of object x has been changed to property (a, w) .

Let $IS = (U, A_{St} \cup A_{Fl} \cup \{d\}, V)$ is a decision table and rules r_1, r_2 have been extracted from IS . The notion of e-action rule was given in [8]. Its definition is recalled below. We assume here that:

- B_{St} is a maximal subset of A_{St} satisfying the condition $r_1/B_{St} = r_2/B_{St}$,
- $d(r_1) = k_1, d(r_2) = k_2$ and $k_1 \leq k_2$,
- $a(r_1) = a(r_2)$, for any $a \in [A_{St} \cap L(r_1) \cap L(r_2)]$,
- $u_i = e_i(r_2)$, for any $e_i \in [A_{St} \cap [L(r_2) - L(r_1)]]$ and $i \leq q$,
- $t_i = c_i(r_2)$, for any $c_i \in [A_{Fl} \cap [L(r_2) - L(r_1)]]$ and $i \leq r$,
- $[v_i = b_i(r_1)] \& [w_i = b_i(r_2)]$, for any $b_i \in [A_{Fl} \cap L(r_1) \cap L(r_2)]$ and $i \leq p$.

Let $F = A_{St} \cap L(r_1) \cap L(r_2)$ and $G = \prod \{a = a(r_1) : a \in F\}$. By (r_1, r_2) -e-action rule on $x \in U$ we mean the expression r :

$$[G \wedge (e_1 = u_1) \wedge (e_2 = u_2) \wedge \dots \wedge (e_q = u_q) \wedge (b_1, v_1 \rightarrow w_1) \wedge (b_2, v_2 \rightarrow w_2) \wedge \dots \wedge (b_p, v_p \rightarrow w_p) \wedge (c_1, \rightarrow t_1) \wedge (c_2, \rightarrow t_2) \wedge \dots \wedge (c_r, \rightarrow t_r)](x) \rightarrow [(d, k_1 \rightarrow k_2)](x)$$

Object $x \in U$ supports (r_1, r_2) -e-action rule r in $IS = (U, A_{St} \cup A_{Fl} \cup \{d\}, V)$, if there exists $y \in U$ and the following conditions are satisfied:

- if $L(r) = \{b_1, b_2, \dots, b_p\}$, then $(\forall i \leq p)[b_i(x) = v_i] \wedge [d(x) = k_1] \wedge [b_i(y) = w_i] \wedge [d(y) = k_2]$
- if $A_{St} \cap L(r_2) = \{a_1, a_2, \dots, a_q\}$, then $(\forall j \leq q)[a_j(x) = u_j] \wedge [a_j(y) = u_j]$
- object x supports rule r_1
- object y supports rule r_2

By the support of e-action rule r in IS , denoted by $Sup_{IS}(r)$, we mean the set of all objects in IS supporting r . In other words, $Sup_{IS}(r)$ contains objects in U having the property $G \wedge (e_1 = u_1) \wedge (e_2 = u_2) \wedge \dots \wedge (e_p = u_p) \wedge (b_1 = v_1) \wedge (b_2 = v_2) \wedge \dots \wedge (b_p = v_p) \wedge (d = k_1)$. By the confidence of r in IS , denoted by $Conf_{IS}(r)$, we mean $[Sup_{IS}(r)/Sup_{IS}(L(r))] \cdot [Conf(r_2)]$.

3 Algorithm DEAR_3

DEAR_3 is a system for producing action rules from data with incomplete values. It constructs a partition-tree, a support-tree, and an action-tree from a set of input samples. DEAR_3 works as follows: extracts classification rules, analyzes them, and finally constructs action rules from certain pairs of classification rules. These rules can be used not only for evaluating discovered patterns but also for reclassifying certain objects from one state into another more preferable state. As mentioned before, algorithm DEAR_3 is the extension of algorithm DEAR_2. Its main difference is in the method used for discovering classification rules. Hence, in this paper, we emphasize on classification rules generation and brief e-action rules construction.

Phase 1: Classification Rules Generation

For classification rule generation, CID [9] algorithm is utilized. It is based on a roulette-wheel technique. The first step of this process is based on the assumption that certain weights are assigned to all attribute values and a missing value is imputed based on a roulette-wheel technique which assigns two most promising values. Individuals having higher frequency are more likely to be selected. Then, a partition-tree and a support-tree is constructed to build classification rules. We assume here that the minimum support is 1 and the minimum confidence is 65%.

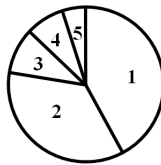


Fig. 1. A roulette wheel with 5 slices. The size of each slice corresponds to the frequency of the individual value of the attribute e .

Step 1: Weighting objects and replacing null values by picking up two promising values using a roulette-wheel technique. Suppose that we have five distinct values for an attribute e such as 1, 2, 3, 4, 5, and their corresponding frequencies are 2.95, 2.47, 0.67, 0.58, and 0.33 respectively. The frequency corresponding to v_i is defined as the weighted number of objects in IS having property (a, v_i) . Each value v_i of the attribute a has a slice of the roulette wheel

allocated in proportion to their frequency (see figure 1). How many times we spin the wheel is determined by the number of distinct values of an attribute. Therefore, in our case, we spin the wheel five times and keep a tally of the result. Let us assume that for object x_3 , values 1, 2, 3, 4, and 5 are chosen 4, 1, 0, 0, and 0 times, respectively. Then, we chose attribute values 1 and 2 to replace a null value in the attribute e for object x_3 . The weight of the selected value 1 is $0.8 : 4/(1 + 4)$, and the weight of the selected value 2 is $0.2 : 1/(1 + 4)$. For every null value, we follow the same process to construct a two-element set of weighted attribute values and use it for replacing that missing value. For perfect knowledge about the state of an object, we assign a 100% weight to it. After imputing all missing values and weighting all attribute values, our incomplete decision system IS is shown in Table 2.

Table 2. An Incomplete Decision System (IS)

U	e	b	c	d
x_1	$\{(4, 0.33), (2, 0.67)\}$	$(1, 1)$	$\{(1, 0.25), (2, 0.75)\}$	$(L, 1)$
x_2	$\{(4, 0.25), (1, 0.75)\}$	$\{(1, 0.5), (5, 0.5)\}$	$\{(1, 0.33), (4, 0.67)\}$	$(L, 1)$
x_3	$\{(1, 0.8), (2, 0.2)\}$	$\{(1, 0.5), (2, 0.5)\}$	$(1, 1)$	$(L, 1)$
x_4	$(1, 1)$	$(3, 1)$	$\{(1, 0.5), (3, 0.5)\}$	$(H, 1)$
x_5	$\{(3, 0.67), (5, 0.33)\}$	$\{(3, 0.75), (4, 0.25)\}$	$(1, 1)$	$(H, 1)$
x_6	$(2, 1)$	$\{(1, 0.9), (6, 0.1)\}$	$(2, 1)$	$(L, 1)$
x_7	$\{(1, 0.6), (2, 0.4)\}$	$\{(1, 0.33), (4, 0.67)\}$	$\{(1, 0.25), (2, 0.75)\}$	$(H, 1)$
x_8	$(1, 1)$	$(3, 1)$	$(2, 1)$	$(H, 1)$
x_9	$\{(2, 0.8), (1, 0.2)\}$	$(1, 1)$	$\{(1, 0.5), (2, 0.5)\}$	$(L, 1)$

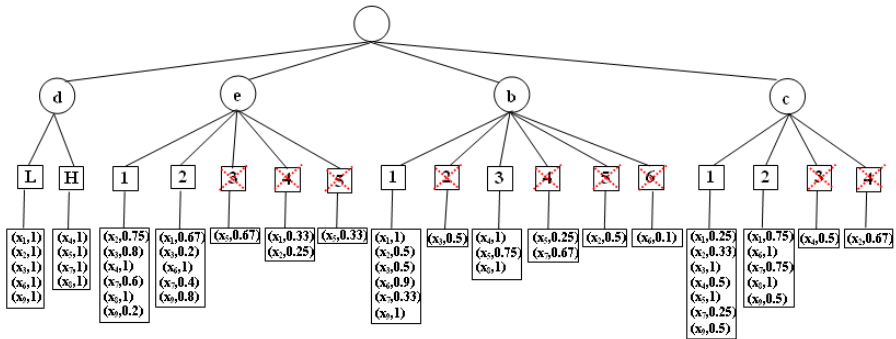


Fig. 2. Partition-Tree

Step 2: **Construction of a Partition-tree** is a process that initially creates granules corresponding to each attribute of the decision system. It partitions the set of objects into weighted granules with respect to each value of each attribute and calculates the support of each granule. If a granule $CLASS_{ISa}[i]$ is below the minimum support, then it is marked as negative. When a granule has negative mark, then it is not considered in later steps. It consists of three components. First one: creates a tree with a dummy root node. The outgoing branches of the root correspond to all possible attributes of the mined data set. Second one: for

each value of an attribute a branch is created with a corresponding set of samples assigned to the new node. Third one: evaluates the support of each leaf node. If the leaf node does not satisfy the threshold, the mark is placed. Referring back to our example, a simple partition-tree with four input attributes e, b, c, d is given in Fig.2. Nine cross marks are placed on the leaf nodes $e = 3, e = 4, e = 5, b = 2, b = 4, b = 5, b = 6, c = 3, c = 4$ and seen as negative marks.

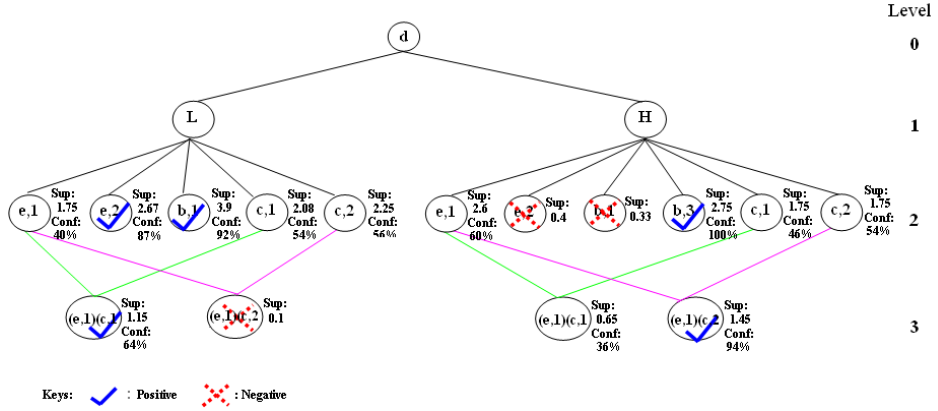


Fig. 3. Support-Tree

Step 3: **Building a support-tree** to group values of classification attributes for each decision value. Create a tree structure that has a decision attribute as the root and its values labelled as its children. Expand the tree by adding relevant grandchild nodes for every child node and label them by the values of the corresponding classification attributes. Let us consider our example again, the root node is a decision attribute d and the two child nodes are labelled by values $\{L, H\}$ of attribute d . When the decision value is L , the values of the classification attributes including $(e, 1)$, $(e, 2)$, $(b, 1)$, $(c, 1)$, and $(c, 2)$ are used. Hence, we place five nodes under node (d, L) . For attribute e , values 1 and 2 are involved; for attribute b , only one value of its domain is used; for attribute c , values 1 and 2 are relevant to decision value L . Using the same strategy we add the child nodes and grandchild nodes of the node (d, H) (see Fig.3). This tree is very useful to determine each $CLASS_{ISa}[i]$ for a given decision value $d[m]$ and it is easy to build.

Step 4: **Evaluate inclusion relation.** The relationship between each classification granule, $CLASS_{ISa}[i]$, and a decision granule, $CLASS_{ISd}[m]$ is checked in this step. If the support of $CLASS_{ISa}[i]$ is below the threshold, it is marked as negative. Otherwise, the support of $a[i] \rightarrow d[m]$ is evaluated; if it is below the threshold, it is marked as negative. Otherwise, the confidence of $a[i] \rightarrow d[m]$ is evaluated; if the condition holds, it is marked as positive. When a granule is marked as negative it means that the corresponding relationship $CLASS_{ISa}[i] \subseteq CLASS_{ISd}[m]$ does not hold and it is not considered any further. The positive

mark means that the rule $a[i] \rightarrow d[m]$ is approved and the corresponding relationship $CLASS_{IS}a[i] \subseteq CLASS_{IS}d[m]$ holds. The computation of support and confidence is based the support-tree and partition-tree. Referring back to our example, for branch (d, L) , the node $(e, 1)$ represents the relationship between $CLASS_{IS}(e, 1)$ and $CLASS_{IS}(d, L)$. The computation of the support and confidence of the inclusion relation are illustrated below.

$$CLASS_{IS}[(e, 1) * (d, L)] = \{(x_2, (0.75 \times 1)), (x_3, (0.8 \times 1)), (x_9, (0.2 \times 1))\} \\ (e, 1) \rightarrow (d, L); (\text{sup} = 1.75 \geq 1; \text{conf} = 1.75/4.35 < 0.62)$$

Since, the support of $CLASS_{IS}[(e, 1) * (d, L)]$ is above the corresponding threshold, its confidence has to be computed as well. But the confidence is below the threshold, so no mark is placed on this node. It means that this node will be considered for extension later on. For all other nodes at the first level of the tree, we follow the same process to compute the support and confidence of the inclusion relation, and also to generate the corresponding classification rule. Going back to our example, the inclusion relation of $CLASS_{IS}(e, 2) \subseteq CLASS_{IS}(d, L)$ holds, since its support is 2.67 and its confidence is 87%. A check (see Figure 3) is placed as a positive mark on the node $(e, 2)$ and the classification rule $(e, 2) \rightarrow (d, L)$ is generated. The term $(e, 2)$ is not extended further to guarantee that the discovered classification rules are the simplest. For the same reason, a positive mark is placed on node $(b, 1)$ and node $(b, 3)$ for the branch (d, L) and (d, H) , respectively. For the branch (d, H) , a cross (see Figure 3) is placed on the node $(e, 2)$ as a negative mark since the support of the inclusion relation $CLASS_{IS}(a, 2) \subseteq CLASS_{IS}(d, H)$ is below the threshold. At the first level of the tree, the following classification rules are generated: $r_1 : (e, 2) \rightarrow (d, L)$; $r_2 : (b, 1) \rightarrow (d, L)$; $r_3 : (b, 3) \rightarrow (d, H)$.

Step 5: Determine whether to terminate the procedure or to expand the support-tree one level deeper. If all terms are marked at this level of the tree, it means that the tree cannot be extended any further and the process has to be terminated. Otherwise, we extend the support-tree one level deeper to form longer terms from unmarked nodes that belong to different attributes under same decision value. Then, go to **step 4** to exam inclusion relation. Going back to our example, we can observe that because the nodes $(e, 1)$, $(c, 1)$, $(c, 2)$ are unmarked in (d, L) -subtree, two grandchild nodes $((e, 1)(c, 1))$ and $((e, 1)(c, 2))$ are added. For the same reason, two grandchild nodes $((e, 1)(c, 1))$ and $((e, 1)(c, 2))$ are added at (d, H) sub-tree. For each newly-formed term, we take the corresponding granules from the partition-tree and intersect them to build a finer granule. This is done by using two recursive functions to call each other in order to dynamically generate finer granules with two or more elements. The computation of the support and confidence of an inclusion relation for a two-element term is illustrated in the following example, node $(e, 1) * (c, 1)$.

$$CLASS_{IS}((e, 1) * (c, 1)) = \{(x_2, 0.755 \times 0.33 = 0.2475), (x_3, 0.8 \times 1 = 0.8), (x_4, 1 \times 0.5 = 0.5), (x_7, 0.6 \times 0.25 = 0.15), (x_9, 0.2 \times 0.5 = 0.1)\}; \\ CLASS_{IS}((e, 1) * (c, 1) * (d, L)) = \{(x_2, 0.2475 \times 1), (x_3, 0.8 \times 1), (x_9, 0.1 \times 1)\}; \\ [(e, 1) \wedge (c, 1)] \rightarrow (d, L); (\text{sup} = 1.475 \geq 1; \text{conf} = 1.475/1.7975 < 0.62)$$

For each node at the second level of the tree, the same method is utilized to calculate its support and confidence. For $(d, L) - subtree$, a negative mark is placed on the node $((e, 1)(c, 2))$ and a positive mark is placed on node $((e, 1)(c, 1))$; for $(d, H) - subtree$, a check is marked on node $((e, 1)(c, 2))$ as shown in Fig. 3. Since only one node is unmarked in each subtree, we can not expand the tree any further. It can be easily checked that the following two optimal classification rules are discovered at the second level of the tree: $r_4 : (e, 1) \wedge (c, 1) \rightarrow (d, L)$, $r_5 : (e, 1) \wedge (c, 2) \rightarrow (d, H)$.

After evaluating inclusion relation, if terms remain unmarked at current level of the tree, we have to repeat step 4 and step 5. Otherwise, we are ready to extract e-action rules.

Phase 2: Generate E-Action Rules

After forming classification rules, action-tree algorithm [5] is utilized to construct e-action rules. There are two basic steps: building the action tree and generating rules. An action tree consists of nodes where attributes are tested. The outgoing branches of a node correspond to all possible outcomes of the test assigned to that node. The construction of an action tree starts with the set of classification rules R at the root of the tree similar to table T_1 in Fig. 4. An attribute with the smallest number of states among all untested stable attributes is selected to partition the rules. For each value of the attribute a branch is created, and the corresponding tuples that have the attribute value specified by that branch are moved to the newly created child node. The algorithm is applied recursively to each child node until all stable attributes are tested. After putting all stable attributes on the tree, the tree is split based on the value of the decision attribute and ready for rule generation. Let us consider our example again and demonstrate the algorithm.

Assume now that our goal is to re-classify objects from the class $d^{-1}(\{d_i\})$ into a new class $d^{-1}(\{d_j\})$. In the example, we assume that $d_i = (d, L)$ and $d_j = (d, H)$. Additional assumption is that the minimum support is 1 and the minimum confidence is 60%.

The construction of an action tree starts with the set R representing the root of the tree (T_1 in Fig. 4). There is only one stable attribute e and two values involved, so we use attribute e to expand the tree one more level. Two branches are created and labelled by the values 1 and 2 of attribute e . All rules in the sub-table T_2 do not contain any stable attributes but they contain different decision values and different values of flexible attributes so now T_2 is divided based on decision attribute d into two new sub-tables T_4 and T_5 . Each leaf represents a set of rules which do not contradict on stable attributes and also define decision value d_i . For the same reason, the sub-table T_3 is split into two sub-tables T_6 and T_7 . Each leaf represents a set of rules, which do not contradict on stable attributes and also define decision value d_i .

The path from the root to that leaf gives the description of objects supported by these rules. When we follow the path labelled by value $[e = 1]$ and $[d = L]$, we get table T_4 . Then, by following the path labelled by $[e = 1]$ and $[d = H]$,

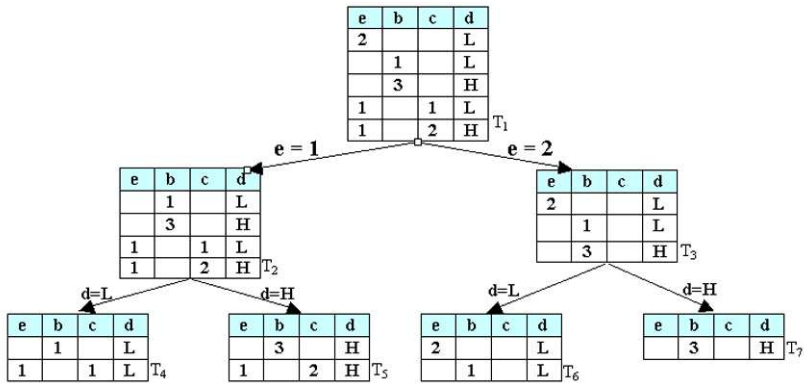


Fig. 4. Action-tree

we get table T_5 . Now, we can compare pairs of rules belonging to these two resulting tables and generate two e-action rules: $[[b, 1 \rightarrow 3]] \Rightarrow (d, L \rightarrow H)$, and $[[(e, 1) \wedge (c, 1 \rightarrow 2)] \Rightarrow (d, L \rightarrow H)]$. For sub-tables T_6 and T_7 , we discover one rule: $[[(b, 1 \rightarrow 3)] \Rightarrow (d, L \rightarrow H)]$. After the rules are formed, we evaluate them by checking their support and confidence. We have discovered two strong e-action rules such as:

$$[[b, 1 \rightarrow 3]] \Rightarrow (d, L \rightarrow H), \text{ sup: } 3.9, \text{ conf: } 92.2\% \text{ and}$$

$$[[(e, 1) \wedge (c, 1 \rightarrow 2)] \Rightarrow (d, L \rightarrow H)], \text{ sup: } 1.1475, \text{ conf: } 60.16\%.$$

The new algorithm (called DEAR₃) was implemented and tested on several datasets including "AdultIncomplete" (AI) and "BreastCancerIncomplete" (BCI). These two datasets are taken from the revised version of the original "Adult" and "BreastCancer". Their null values are replaced by using roulette wheel strategy to pick 2 values from a corresponding domain and assigning weight from $[0,1]$ to them. The original tables are available at [http://www.cs.toronto.edu/~delve]. The first one has 2000 records described by 14 attributes. Only four attributes *age*, *NTVC-NTRY*, *race*, and *sex* are stable. The second one has 191 objects describe by 10 features. There is only one stable attribute *age*. The tested results are presented in Table 3 to show the time needed for DEAR₃ to extract rules and then form e-action rules from these two datasets.

Table 3. Time need to extract classification rules and E-action Rules

Dataset	Min. Sup.		Min. Conf.		Classificationrules		E – ActionRules	
					No	Time	No	Time
AI	10	90%	165	7min55sec	73	54sec		
AI	20	95%	70	4min51sec	3	3sec		
BCI	5	85%	25	7sec	18	0sec		
BCI	5	75%	50	3sec	54	0sec		
BCI	3	75%	41	10sec	63	1sec		

4 Conclusion

E-action rules are essential for automatic evaluation of discovered classification rules and formation of workable strategies. These strategies can be turned into actions and these actions can ultimately benefit the users. Most of the collected real-world data sets are incomplete. In this paper we presented a novel algorithm DEAR_3 for extracting actionable knowledge from such data. This algorithm was implemented as System DEAR_3 which is a significant improvement of our previous system DEAR_2 [7].

References

1. Pawlak Z (1981) Information systems - theoretical foundations. In: Information Systems Journal, Vol. 6, 205–218
2. Pawlak Z (1985) Rough Sets and Decision Tables. In: Lecture Notes in Computer Science 208, Springer-Verlag, 186–196
3. Raś Z, Wieczorkowska A (2000) Action rules: how to increase profit of a company. Proceedings of PKDD'00, Lyon, France, LNCS/LNAI, No. 1910, Springer-Verlag, 587–592
4. Raś Z W, Tsay L-S (2003) Discovering extended action-rules (System DEAR). In: Proceedings of the IIS'2003 Symposium, Zakopane, Poland, Advances in Soft Computing, Springer-Verlag, 293–300
5. Raś Z W, Tsay L-S, A. Dardzinska (2005) Mining E-Action Rules. In: Mining Complex Data, Proceedings of 2005 IEEE ICDM Workshop in Houston, Texas, Published by Math. Dept., Saint Mary's Univ., Nova Scotia, Canada, 2005, 85-90
6. Tsay L-S, Raś Z W, and A. Wieczorkowska (2004) Tree-based algorithm for discovering extended action-rules (System DEAR2). In: Proceedings of the IIS'2004 Symposium, Zakopane, Poland, Springer, 459-464
7. Tsay L-S, Raś Z W (2005) Action rules discovery: System DEAR2, method and experiments. Journal of Experimental and Theoretical Artificial Intelligence, Taylor Francis, Vol. 17, No. 1-2, 119-128
8. Tsay L-S, Raś Z W (2006) E-Action Rules. In: Foundations of Data Mining, Studies in Computational Intelligence, Springer, 2006, will appear
9. Tsay L-S, Raś Z W (2006) Discovering E-Action Rules from Incomplete Information Systems. In: Proceedings of IEEE International Conference on Granular Computing (IEEE GrC 2006), May 10-12, 2006, Atlanta, Georgia, 518-521

Mining and Modeling Database User Access Patterns

Qingsong Yao, Aijun An, and Xiangji Huang*

Department of Computer Science and Engineering, York University, Toronto, M3J 1P3 ,Canada
{qingsong, aan}@cs.yorku.ca, jhuang@yorku.ca

Abstract. We present our approach to mining and modeling the behavior of database users. In particular, we propose graphic models to capture the database user's dynamic behavior and focus on applying data mining techniques to the problem of mining and modeling database user behaviors from database trace logs. The experimental results show that our approach can discover and model user behaviors successfully.

1 Introduction

Workload analysis has played an important role in optimizing the performance of database systems. While most work on database workload analysis focuses on providing statistical summaries and run-time behavior on the physical resource level of the database system, it has been brought into attention that analysis of task-oriented user sessions provides useful insight into the query behavior of the database users [7,8]. A session is a sequence of queries issued by a user (or an application) to achieve a certain task. It consists of one or more database transactions, which are in turn a sequence of operations performed as a logical unit of work. Analysis of sessions allows us to discover high-level patterns that stem from the structure of the task the user is solving.

In this paper, we first describe our approach to modeling database user behaviors. We use the concept of *user access event* and *user access graph* to represent the database queries and the dynamic relationship among the queries. Then, we describe how we apply data mining techniques to mine database users' access patterns from database traces. Our contributions in this paper are summarized as follows. First, we present a complete process for mining user access patterns from database trace logs, including data preprocessing, session identification, session clustering and the generation of user access graphs. Second, we present the use of Markov models to build the user access graphs, and provide experimental results showing that the discovered user access graphs can model the database user behavior well.

2 User Access Graph

The SQL queries submitted by a database user are not random. They follow business rules or logics. We use *user access event* and *user access graph* to represent the query

* This work is supported by Communications and Information Technology Ontario (CITO) and the Natural Sciences and Engineering Research Council of Canada (NSERC).

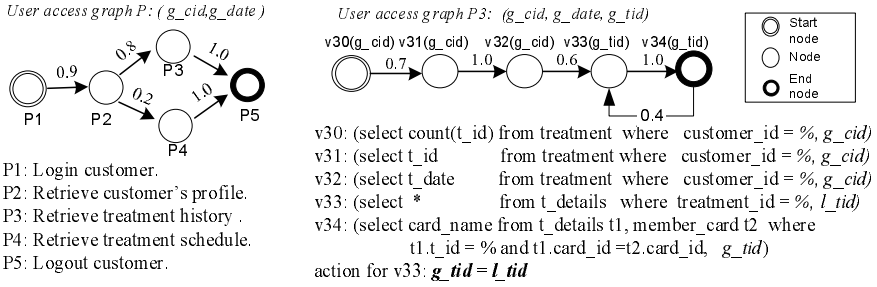


Fig. 1. An example of user access graphs

Table 1. An instance of treatment-history-retrieving procedure

ID	Query	Business meaning
q30	select count(t_id) from treatment where customer_id = 'c101'	get customer c101's treatment count
q31	select t_id from treatment where customer_id = 'c101'	get a list of treatment id
q32	select t_date from treatment where customer_id = 'c101'	get a list of treatment date.
q33	select * from treatment_details where treatment_id = 't202'	get the details of one treatment
q34	select * from treatment_payment where treatment_id = 't202'	get the payment information of one treatment

format and the query execution order, respectively. Given an SQL query, we transform it into two parts: an *SQL template* and a set of *parameters*. We treat each data value embedded in the query as a parameter, and the SQL template is obtained by replacing each data value with a wildcard character (%). In many situations, the SQL queries submitted by a user have the same SQL template and are only different in data values. In this paper, we use a *user access event* to represent queries with similar format. A user access event contains an SQL template and a set of parameters. For example, the SQL query “select name from customer where id = 'c101'” can be represented by event (“select name from customer where id = '%’”, g_cid), where g_cid is a parameter.

Given a set of SQL query sequences that have similar formats and execution orders, we use a *user access graph* to represent the query execution order. A user access graph is a directed graph that has one start node and one or more end nodes. Each node in the graph is a user access event or a user access graph. Each node v_i has a support value ρ_{v_i} , which is the occurrence frequency within the set of user sessions, and an edge is represented by $e_k : (v_i, v_j, \sigma_{v_i \rightarrow v_j})$, where $\sigma_{v_i \rightarrow v_j}$ is the probability of v_j following v_i , which is called the confidence of the edge. There are two types of user access graphs depending on the granularity of the nodes. The first type is called the basic user access graph, whose nodes represent only events. The second type is a high level user access graph, in which each node is represented by a basic user access graph. A high level user access graph is used to describe the execution orders among sessions that are performed by the same user. For example, in a client application, there is a patient-information model that helps an employee to retrieve a patient's treatment history and treatment schedule. When a user logs in, the corresponding profile is retrieved. Then he/she can either retrieve the treatment history information or treatment schedule. Figure 1 shows two user access graphs for the patient-information model. Graph P, illustrated on the left side, is a high-level user access graph that describes the execution orders of five

sub-models. 80% of the users retrieve treatment history information (since $\sigma_{P2 \rightarrow P3} = 0.8$). Graph $P3$, illustrated on the right side, is a basic user access graph that represents the treatment-history retrieval sub-model. Table 1 shows an instance of the treatment-history sub-model that contains four consecutive SQL queries.

A parameter in an event can be a constant, a dependent variable or an independent variable. The value of a constant parameter cannot be changed by any of the events in the user access graph. An independent variable can take different values and its value does not depend on any other variable or constant in the graph. The value of a dependent variable can be changed by an event and its value depends on some other variables in the graph. For example, g_cid is a constant parameter in the user access graph $P3$ on the right side of Figure 1, since its value does not change by any of the events on the graph. g_tid in event $v34$ is a dependent variable because its value is set by event $v33$ to be the value of parameter l_tid in $v33$, which is an independent variable. From Table 1, we observe that query $q34$ can not be anticipated until query $q33$ is submitted. Thus, the corresponding user access event $v34$ is determined by $v33$. We call events whose parameters are not independent variables *determined events* since the corresponding queries are known before they are submitted. Other events are called *undetermined events*. For example, event $v33$ is undetermined event, and $v31$, $v32$, and $v34$ are determined event. A determined event can be *result-determined* by the results of other events, *parameter-determined* by the parameters of other events, or it only depends on constants or global constant which is called a *graph-determined* event. A graph-determined event does not depend on any other events. The difference between an undetermined event and a result-determined event is that the parameters of the latter can be derived from query result. For example, $v31$, $v32$ are graph-determined event, and $v34$ is parameter-determined by $v33$.

3 Discovering User Access Graphs

Our objective is to mine user access graphs from database traces. The procedure of mining user access graphs includes the following steps. First, database traces are collected and preprocessed, in which noisy and irrelevant data are removed and the log entry is transformed to a meaningful format. Second, every SQL query in the traces is transformed into an *SQL template* and a set of *parameters*. We treat a data value in a given SQL query as a parameter, and the SQL template can be obtained by replacing each parameter in the query with a wildcard character '%'. A *user access event* is assigned to each SQL template to represent a set of *similar queries*. By replacing the SQL statements in the log entry with the corresponding user access events, we obtain sequences of user access events, referred to as *event sequences*. In the third step, we apply a session identification method to detect session boundaries from an event sequence which contains multiple sessions submitted sequentially [3,10]. After session instances are identified, we further group them into session classes using a clustering method described in [9]. Each session class contains a set of session instances. Finally, for each session class, a user access graph is built. In the next section, we describe the last step that builds a user access graph from a set of session instances of the same class.

4 Modeling Database User Sessions

A session class g contains a set of session instances, $\{s_1, s_2, \dots, s_n\}$. Each session instance is a sequence of requests. The procedure of modeling session classes contains three steps. First, we ignore the relationships among parameters in the requests and consider the request execution order only. In this step, we use Markov models to build the structures of the basic user access graphs, i.e., the nodes and the edges of the graphs, where nodes represent user access events. Second, we find the relationships between the parameters in use access events and introduce variables/actions to the graphs. Finally, to build a high-level user access graph, we replace each session instance s_i with the corresponding session class, and obtain a collection of session class sequences, one for each user. The idea of building low-level user access graphs can be used to build high-level user access graphs.

4.1 User Access Graph Generation with Markov Model

Let $X_1, X_2, \dots, X_n, \dots$ be a sequence of random variables taking values from a finite set of values $S = \{x_1, x_2, \dots, x_n\}$. The random variables are said to form a k^{th} -order Markov chain model if

$$P(X_i | X_1, X_2, \dots, X_{i-1}) = P(X_i | X_{i-k}, \dots, X_{i-2}, X_{i-1}), \quad (1)$$

for all values of i . In other words, the current variable at time i depends on previous k variables. We can think of the values of X_i as *states*, and the Markov model as a finite state process with transitions between states specified by probabilities $P(x_i | x_{i-k}, \dots, x_{i-2}, x_{i-1})$. The probabilities can be represented by an $n^k \times n$ matrix. k is called the order of the model. A k^{th} -order Markov model can always be converted into an equivalent first-order Markov model. We can introduce random variables as $Z_i \doteq X_{i-k+1}, X_{i-k+2}, \dots, X_i$, and the Z process forms a first-order Markov model [4].

A zero-order Markov model makes prediction based on the action reference statistics, and a first-order Markov model makes predictions based on the last action performed by the user. In general, zero-order Markov model and first-order Markov model have limited success in representing the dynamic user behavior since these models do not look far into the past. A k^{th} -order Markov model make predictions by looking at the last k actions performed by the user, which lead to a state-space that contains all possible sequences of k actions. The large number of possible states leads to a large transition probability matrix.

To build a user access graph for each session class, we generate a higher-order Markov model¹ from a group of sessions. We treat each state of the Markov model as a node in the user access graph, and each state transition corresponds to an edge in the graph. Thus, the Markov state transition diagrams describe the structure of a user access graph. The procedure for building a Markov model is straightforward. We first scan

¹ Strictly speaking, the model used is first-order Markov models, where every state is a sequence of k requests. But since it is transformed from a k^{th} -order Markov in which every request corresponds to a state, it is equivalent to a higher-order Markov model.

all session instances once, generate the possible states and the state transition probabilities. Once a session instance $s = \langle r_1, \dots, r_n \rangle$ is retrieved, we add a special request "start" at the beginning of s , and s forms totally $n+1$ states: $(start)$, $(start, r_1), \dots, (r_{n-k}, \dots, r_n)$. The state frequency and state transition frequency are updated correspondingly. In the second step, the states and state transitions are pruned according to the state support threshold (i.e., the minimal support value for the node) and the confidence threshold (i.e., the minimal confidence value for the edge). We refer to such models as the *pruned Markov models*. A corresponding state transition diagram can also be easily constructed from the transition matrix. By using pruned Markov models, the number of states and the number of state transitions are reduced.

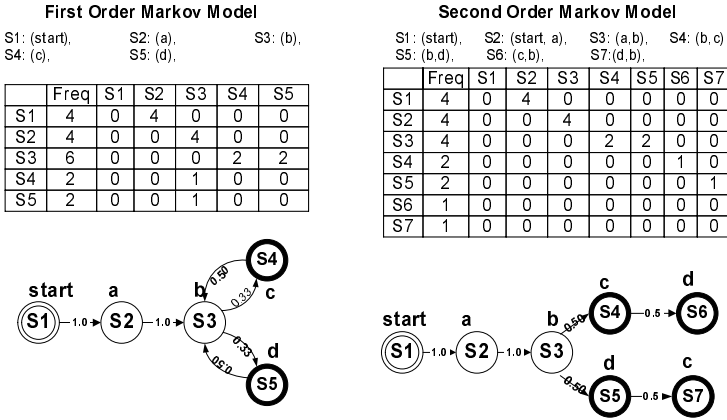


Fig. 2. First-order and second-order pruned-Markov Model

For example, given four session instances $\langle a, b, c \rangle$, $\langle a, b, c, b \rangle$, $\langle a, b, d \rangle$, $\langle a, b, d, b \rangle$, the first-order and second-order pruned-Markov models and their corresponding transition diagrams are illustrated in Figure 2. The confidence threshold is 0.33 and the support count threshold is 1. We observe that first-order model has fewer states than the second-order model, but it cannot distinguish the two requests of b in session $\langle a, b, d, b \rangle$ and $\langle a, b, c, b \rangle$. We can also observe that the number of states in a pruned second-order models is not large (compared with $4^3 = 64$ in the un-pruned model), and it can model the dynamic behavior of the session class well.

4.2 Finding Relationship Between Nodes

The relationship discovery problem is defined as follows. Given a user access graph G_s and a set s of sessions $\{s_1, \dots, s_n\}$ belonging to it, find the relationships between the node parameters in G_s , and introduce constants, variables and edge actions into G according to the relationships. In this step, each request in a session is an SQL query and can be represented by using an user access event and a set of values. Each data value corresponds to a parameter of the event. Meanwhile, each request can be mapped

into a node in the user access graph. The relationships between node parameters can be discovered by analyzing the data values between the requests.

We assume that all parameters in the nodes of G make a virtual relation R , where each parameter is one attribute of R . Each session s_i , which contains a sequence of queries, corresponds to a tuple of R . The data values in all the queries in s_i are the attribute values of that tuple. Therefore, the set s of sessions make an instance of relation R , referred to as r . Thus, the problem of finding parameter relationships of a user access graph G_s with sessions s becomes the problem of finding functional dependencies and dependency functions in an instance r of relation R .

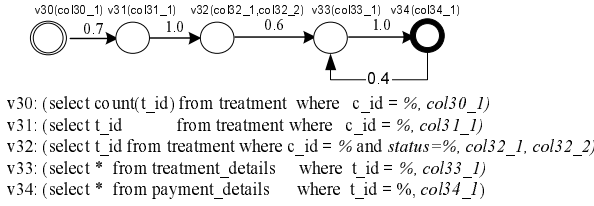


Fig. 3. Original user access graphs

Suppose a user access graph listed in Figure 3 is obtained. The graph contains five user access events where each event contains a set of parameters. we first assign a unique name to each parameter. The corresponding virtual relation contains six attributes, referred to as $col30\_1$, $col31\_1$, $col32\_1$, $col32\_2$, $col33\_1$, $col34\_1$, respectively. Table 1 shows a session that contains five consecutive queries. The corresponding tuple of R is illustrated in Table 2, which is obtained by extracting the data values from the queries, and mapping them to the corresponding attributes.

Table 2. A record of relation R

col30_1	col31_1	col32_1	col32_2	col33_1	col34_1
'c101'	'c101'	'c101'	'1'	't202'	't202'

From the virtual relation instance, we can find the functional dependencies. The parameter relationship discovery procedure usually contains the following steps:

1. Virtual relation and relation instance generation. In this step, virtual relation R and relation instance r are generated from sessions s and user access graph G_s .
2. Functional dependency inference. In this step, functional dependencies are inferred from r by using a functional dependency inference algorithm. Two kinds of special dependencies, $\{\rightarrow Y\}$, and $\{X \leftrightarrow Y\}$, are two common cases in our inference problem, which can be found in an efficient way. Dependency $\{\rightarrow Y\}$ means that the value of Y is a constant and will not change. It can be discovered by examining the cardinality of Y , which is the number of distinct values of attribute Y in the relational instance. Dependency $\{X \leftrightarrow Y\}$ means that X and Y agree on all possible values.

3. Dependency function discovery. In this step, dependency functions are discovered from relation instance r . A dependency function, for a given functional dependency $X \rightarrow Y$, can help to obtain a value of Y by giving the value of X . When X and Y take numeric values, we use regression analysis to find a function. When the target attribute Y is categorical, we can treat the relation instance r as the training data, and each tuple in r has a class label y that is the corresponding value of attribute Y . Thus, a set of classification rules can be obtained from the training data. The classification rules can be used to describe the relationship among query parameters, and can be viewed as the dependency functions.
4. Graph variable and edge action introduction. In this step, graph variables and edge actions are introduced according to the dependencies and functions discovered in step 2 and 3.

5 Experimental Results

We evaluated our modeling method on a clinic OLTP application. The clinic is a private physiotherapy clinic located in Toronto. It has five branches across the city. It provides services such as joint and spinal manipulation and mobilization, post-operative rehabilitation, personal exercise programs and exercise classes, massage and acupuncture. The client program consists of several models or components. Each model consists of sequences of SQL statements, and the query execution order is controlled by the business logics embedded in the model. The execution of these models in an instance of the client program also has certain rules. Each day, the client applications installed in the branches make connections to the center database server, which is Microsoft SQL Server 7.0. In each connection, a user may perform one or more tasks, such as checking in patients, making appointments, displaying treatment schedules, explaining treatment procedures and selling products. The trace file is collected by using Microsoft SQL Profiler². The SQL Profiler can monitor all events that occur in the SQL Server database, and a user can define the events to be monitored manually. We use the SQL Profiler to collect all SQL queries submitted by the client application within a period of observation time. The database trace log (400M bytes) contains 81,417 events belonging to 9 different applications, such as front-end sales, daily report, monthly report, data backup, and system administration. The target application of the paper is the front-end sales application. After preprocessing the trace log, we obtain 7,244 SQL queries, 18 database connection instances of the front-end sales application. 2989 types of queries are found from the trace log, and only 4% of the queries have a frequency over 5. The queries are classified into 190 user access events, which have an average frequency of 38. 18 user access event sequences are obtained by replacing the queries with the corresponding user access events, each sequence corresponds to one client application instance.

We choose half of the database traces as the training data, and randomly select four user access event sequences from the remaining traces as the test data. Session identification is first performed on the data sets and then the sessions in the training data are clustered using our session clustering method into 21 session clusters (classes). We

² SQL Profiler is registered trademark of Microsoft Corporation.

prune out the clusters that have less than 10 session instances. We construct the corresponding user access graphs for the remaining session clusters by using the algorithm proposed in Section 4. In the experiment, we use the minimal support value of 3 and the minimal confidence value of 0.05. Representative user access graphs are shown in Figure 4. User access graph *P1_1*, *P1_2* and *P1_3* are high-level graphs. An instance of user access graph *P1* which retrieves a given customer's profile is listed in Table 3. The graph instance can be interpreted as employee '1025' retrieve customer '1074's profile and treatment schedule at branch 'scar' on 2003-03-04. Thus, the graph has four parameters: user id (*g_uid*), customer id (*g_cid*), branch id (*g_bid*), and login date (*g_date*), where *g_date* is a constant. We observe that once query q9 is submitted, the format of query q10, q20 and q49 are determined, thus they can be predicted. In the correspond graph *P1*, events *v30*, *v9*, *v47* are un-determined event, and events *v10*, *v20* and *v49* are parameter-determined by *v9*.

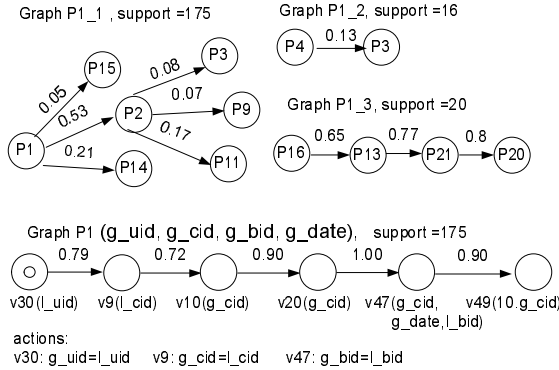


Fig. 4. User access graphs

Table 3. An instance of user access path P1

ID	Statement
q30	select authority from <i>employee</i> where employee_id ='1025'
q9	select count(*) as num from <i>customer</i> where cust_num = '1074'
q10	select card_name from customer t1,member_card t2 where 1.cust_num = '1074' and t1.card_id = t2.card_id
q20	select contact_last,contact_first from <i>customer</i> where cust_num = '1074'
q47	select t1.branch,t2.* from <i>record</i> t1, <i>treatment</i> t2 where t1.contract_no = t2.contract_no and t1.cust_id ='1074' and check_in_date = '2003/03/04' and t1.branch = 'scar'
q49	select top 10 contract_no from <i>treatment_schedule</i> where cust_id = '1074' order by checkin_date desc

To evaluate the user access graphs, we use them to predict the next request from a sequence of the request history $x : (x_1, x_2, \dots, x_n)$, where x is part of a session instance in the test data. The method for predicting next request is straightforward. for each session class g_i , the probability that a request y_i be the next request under g_i is:

$$P(y_i|x, g_i) = P(y_i|x_1x_2\dots x_n, g_i) = P(y_i|x_{n-k+1}\dots x_n, g_i)$$

where $P(y_i|x_{n-k+1}...x_n, g_i)$ can be obtained from the transition probabilities of g_i . We predict next request to be y_i only when $P(y_i|x, g_i)$ exceeds a certain threshold and $P(y_i|x, g_i)$ achieves the maximum value among all classes. If the next request is predicted correctly, we said it is a "match". We use *F-Measure* to measure the prediction performance, which is defined as follows:

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall},$$

where *precision* is defined as the ratio of the number of "matches" to the total number of predicted requests and the *recall* is the hit-rate, which is the portion of the requests that are correctly "matched".

The result for request prediction is shown in Table 4, where there are 5,490 requests in the test data. The prediction threshold is 0.8. Table 4 shows that the overall *precision* of the prediction algorithm is very high, but the *recall* is not. The reason is that we can not make prediction when there is limit request histories, which usually occurs when a session just begins. The result shows that our method can effectively represent the users' access pattern and can predict users' next requests well. In this application, a 5-order Markov-model performs the best.

Table 4. Request Prediction Performance

Markov order	predicted requests	matched requests	precision	recall	F-Measure
1	2718	2627	0.97	0.48	0.64
2	2870	2776	0.97	0.51	0.67
3	2874	2791	0.97	0.51	0.67
4	2894	2811	0.97	0.52	0.67
5	2918	2839	0.97	0.52	0.68
6	2927	2848	0.97	0.52	0.68
7	2918	2842	0.97	0.52	0.68
8	2926	2850	0.97	0.52	0.68
9	2377	2345	0.99	0.43	0.60
10	2425	2393	0.99	0.44	0.61
11	1979	1947	0.98	0.36	0.52
12	1979	1947	0.98	0.36	0.52

6 Related Work

Previous workload studies focus on describing the statistical summaries of run-time behavior [7,12,2], clustering database transactions [11,5], predicting the buffer hit ratio [1], and improving caching performance [6,8]. Chaudhuri *et al* [7] suggest to use SQL-like primitives for workload summarization. A few examples of workload summarization and the possible extensions to SQL and the query engine to support these primitives are also discussed in the paper. In [12], a relational database workload analyzer (REDWAR) is developed to characterize the workload in a DB2 environment.

Their study focused on statistical summaries, such as averages, variations, correlations and distributions, and description of the runtime behavior of the workload. In [2], Hsu *et al* analyze the characteristics of the standard workload TPC-C and TPC-D, and examine the characteristics of the production database workloads of ten of the world's largest corporations. Yu *et al* [11] propose an affinity clustering algorithm which partitions the transactions into clusters according to their low-level database reference patterns. Nikolaou *et al* [5] introduce several clustering approaches by which the workload can be partitioned into classes consisting of units of work exhibiting similar characteristics. Dan *et al* [1] analyze the buffer hit probability based on the characterization of low-level database access to physical pages. They make distinctions among three types of access patterns: locality within a transaction, random access by transactions, and sequential accesses by long queries. However, all above mentioned clustering algorithms are based on the low-level reference patterns.

7 Conclusion

We have presented a new approach to mining and modeling database user access patterns. To our knowledge, this is the first attempt to analyze database user access patterns systematically. We use the user access events to represent the static features of a database workload, and use the user access graphs to describe database query execution orders. The mining results from our approach can be used to tune the database system and predict incoming queries based on the queries already submitted, which can be used to improve the database performance by effective query prefetching, query rewriting and cache replacement. Experiments show that our approach provides a promising avenue for mining and modeling database users' access behaviors. The work presented in the paper has a broader impact on the database and data mining fields. It can also be used on Web log analysis and DNA sequence analysis. We believe that our approach can help database vendors develop intelligent database tuning tools and rule-based database gateway. In the future, we plan to apply the proposed ideas to OLAP trace logs.

References

1. Asit Dan, Philip S. Yu, and Jen-Yao Chung. Characterization of database access pattern for analytic prediction of buffer hit probability. *VLDB Journal*, 4(1):127–154, 1995.
2. W. W. Hsu, A. J. Smith, and H. C. Young. Characteristics of production database workloads and the tpc benchmarks. *IBM Systems Journal*, 40(3), 2001.
3. Xiangji Huang, Qingsong Yao, and Aijun An. Applying language modeling to session identification from database trace logs. *Knowledge and Information Systems: An International Journal (KAIS)*, 2006.
4. Frederick Jelinek. *Statistical Methods for Speech Recognition*. The MIT Press, 1998.
5. Christos Nikolaou, Alexandros Labrinidis, Volker Bohn, Donald Ferguson, Michalis Artavanis, Christos Kloukinas, and Manolis Marazakis. The impact of workload clustering on transaction routing. Technical Report TR98-0238, 1998.
6. Carsten Sapia. PROMISE: Predicting query behavior to enable predictive caching strategies for OLAP systems. In *DAWAK*, pages 224–233, 2000.

7. Vivek R. Narasayya Surajit Chaudhuri, Prasanna Ganesan. Primitives for workload summarization and implications for sql. In *VLDB 2003, Berlin, Germany*, pages 730–741, 2003.
8. Qingsong Yao and Aijun An. Using user access patterns for semantic query caching. In *Database and Expert Systems Applications (DEXA)*, 2003.
9. Qingsong Yao, Aijun An, and Xiangji Huang. A distance-based algorithm for clustering database user sessions. In *ISMIS*, 2005.
10. Qingsong Yao, Xiangji Huang, and Aijun An. A machine learning approach to identifying database sessions using unlabeled data. In *DaWaK*, pages 254–264, 2005.
11. P. S. Yu and A. Dan. Performance analysis of affinity clustering on transaction processing coupling architecture. *IEEE TKDE*, 6(5):764–786, 1994.
12. Philip S. Yu, Ming-Syan Chen, Hans-Ulrich Heiss, and Sukho Lee. On workload characterization of relational database environments. *Software Engineering*, 18(4):347–355, 1992.

Belief Revision in the Situation Calculus Without Plausibility Levels

Robert Demolombe¹ and Pilar Pozos Parra²

¹ ONERA Toulouse, France

`Robert.Demolombe@cert.fr`

² Universidad Tecnológica de la Mixteca, Mexico

`pilar@mixteco.utm.mx`

Abstract. The Situation Calculus has been used by Scherl and Levesque to represent beliefs and belief change without modal operators thanks to a predicate plays the role of an accessibility relation. Their approach has been extended by Shapiro et al. to support belief revision. In this extension plausibility levels are assigned to each situation, and the believed propositions are the propositions that are true in all the most plausible accessible situations.

Their solution is quite elegant from a theoretical point of view but the definition of the plausibility assignment, for a given application domain, raises practical problems.

This paper presents a new proposal that does not make use of plausibilities. The idea is to include the knowledge producing actions into the successor state axioms. In this framework each agent may have a different successor state axiom for a given fluent. Then, each agent may have his subjective view of the evolution of the world. Also, agents may know or may not know that a given action has been performed. That is, the actions are not necessarily public.

1 A Brief Introduction to the Situation Calculus

In the very beginning the Situation Calculus was introduced by McCarthy [4]. It has been used by several authors to propose solutions to the frame problem and a simple solution has been proposed by Reiter in [6]. Since several variants of the Situation Calculus have been presented in the literature, we refer to the Situation Calculus as it is defined in Reiter's book [7].

The Situation Calculus is a many sorted classical logic (most of it uses first order logic). There are two kinds of predicates. Those whose truth values may change when actions are performed, which are called “fluents”, and those whose truth values do not change. The fluents have exactly one argument of the type situation which is the last argument. A similar deal is made for function symbols.

For instance, if we consider a fragile toy, the fluent *broken(s)* can be used to represent the fact that in the situation *s* the toy is broken. The fluent *position(x, s)* can represent the fact that in the situation *s* the toy is in the position *x*. If the color of the toy never changes, its color is not represented by a fluent. It is represented by a predicate. For instance, *color(x)* which has no argument of the type situation, and whose meaning is that the toy color is *x*.

The situations are represented by terms that may be constants of the type situation, or complex terms formed with the designated binary function symbol *do*. If *a* is a term of the type action and *s* is of the type situation, then *do(a, s)* is of the type situation. From an intuitive point of view, *do(a, s)* represents the situation resulting from the performance of *a* in the situation *s*.

For example, the constant *s*₀ can be used to represent the initial situation, and, if *drop* and *repair* are of the type action, the terms: *do(drop, s*₀*)*, *do(repair, s*₀*)* and *do(repair, do(drop, s*₀*)**)*, are of the type situation.

The solution to the frame problem is based on the assumption that we (the people who are modeling an application domain) know **all** the actions, and the contexts, that can change the truth value of each fluent.

The fact that the actions *drop* and *repair* cause respectively the fluent *broken(s)* to be true and false is represented by the following axioms:

$$\begin{aligned} \forall s \forall a (a = \textit{drop} \rightarrow \textit{broken}(\textit{do}(a, s))) \\ \forall s \forall a (a = \textit{repair} \rightarrow \neg \textit{broken}(\textit{do}(a, s))) \end{aligned}$$

The fact that there is no other action that can change the truth value of *broken(s)* is represented by the “Successor State Axiom (SSA)”:

$$\forall s \forall a (\textit{broken}(\textit{do}(a, s)) \leftrightarrow a = \textit{drop} \vee \textit{broken}(s) \wedge \neg(a = \textit{repair}))$$

The general form of a SSA for a fluent *p* is¹:

$$(S_p) \quad \forall s \forall a \forall \mathbf{x} (p(\mathbf{x}, \textit{do}(a, s)) \leftrightarrow \Gamma_p^+(\mathbf{x}, a, s) \vee p(\mathbf{x}, s) \wedge \neg \Gamma_p^-(\mathbf{x}, a, s))$$

where $\Gamma_p^+(\mathbf{x}, a, s)$ and $\Gamma_p^-(\mathbf{x}, a, s)$ are first order formulas whose free variables are: $\mathbf{x}$, *a* and *s*.

For instance, for the fluent *position(x, s)* we have:

$$\Gamma_{\textit{position}}^+(\mathbf{x}, a, s) = \exists y (a = \textit{move}(y, x) \wedge \textit{position}(y, s))$$

In [8] Scherl and Levesque have extended the Situation Calculus to represent beliefs and belief changes. One of the basic ideas is to see situations like possible worlds in Kripke models². The second idea is to introduce a fluent *K(s', s)* which has the same intuitive meaning as an accessibility relation in doxastic or epistemic logics.

To represent what an agent believes in a given situation, say *s*₀, Reiter says in [7]: “we can picture this in terms of an agent inhabiting *s*₀, and imagining all possible alternative situations to the one he is in”. Those imaginary situations are related with the real one through the fluent *K*, that is, they are the situations *s'* such that: *K(s', s*₀*)*.

In general, *Knows(φ, s)*³ is used as a notation to represent the fact in the situation *s* an agent believes *φ*. We have:

$$\textit{Knows}(\phi, s) \stackrel{\text{def}}{=} \forall s' (K(s', s) \rightarrow \phi[s'])$$

¹ $\mathbf{x}$ is used as an abbreviation for the tuple of variables $x_1, \dots, x_n$, and $\forall \mathbf{x}$ is an abbreviation for $\forall x_1 \dots \forall x_n$.

² This is an intuitive analogy. In fact there are significant differences between possible worlds and situations. One of them is that situations represent histories.

³ The notation *Knows(φ, s)* is a bit misleading because *φ* is not necessarily true in *s*.

where $\phi[s']$ is the formula obtained by replacing in each fluent the argument of type situation by s' .

To define belief changes **two questions** have to be answered:

- (1) what are the new accessible situations after performance of an action a ?
- (2) what are the truth values of the fluents in the new accessible situations?

The answer to the first question depends on the type of action a . Two different types of actions are considered: those that are “knowledge producing actions” and those that are not. Each knowledge producing action informs the agent about the truth of a given proposition in the situation where the agent is.

For instance, we can have the action *look* that informs the agent about the truth of *broken*(s), and another knowledge producing action to inform him about the toy position.

Given that each knowledge producing action informs about the truth value of a given proposition, after performance of a knowledge producing action, the new accessible situations s'' are the successor of the situations where this proposition has the same truth value as in real situation s . In other words, the imaginary situations that are **inconsistent** with what has been observed are **removed**.

In the case of a non knowledge producing action, the new accessible situations are the successors of the situations that were accessible before to perform a . That is, if s' was accessible from s , after performance of a the corresponding new accessible situation from $do(a, s)$ is $do(a, s')$.

In general, the new accessible situations are defined by the following axiom:

$$(S_K) \quad \forall s \forall s'' \forall a (K(s'', do(a, s)) \leftrightarrow \exists s' (K(s', s) \wedge s'' = do(a, s') \wedge (\\ \neg(a = \alpha_1) \wedge \dots \wedge \neg(a = \alpha_n)) \\ \vee a = \alpha_1 \wedge (\phi_1(s) \leftrightarrow \phi_1(s')) \\ \dots \\ \vee a = \alpha_n \wedge (\phi_n(s) \leftrightarrow \phi_n(s')))))$$

where $\alpha_1, \dots, \alpha_n$ is the set of all the knowledge producing actions, and $\phi_1, \dots, \phi_n$ are their corresponding propositions.

The answer to the second question is that the truth values of the fluents in the new imaginary accessible situations are determined from the previous ones by the SSAs. For example, if we have $K(do(a, s'), do(a, s))$, the truth value of *broken*($do(a, s')$) is determined by the SSA of the fluent *broken*, and by the truth value of *broken*(s'). Then, for non knowledge producing actions we can say that, in some sense, the beliefs change in the same way the world does.

However, this formalization of belief change raises a big problem in the case of revision because if every imaginary situation is inconsistent with the situation s , they are all removed. The result is that in $do(a, s)$ every proposition is believed, and we have inconsistent beliefs.

To remedy this problem Shapiro et al. have proposed in [9] to assign plausibility levels to all the situations, and have defined the believed propositions as the propositions which are true in all the most plausible accessible situations.

However, this new belief definition raises practical problems when we have to define the plausibility function pl for a given application domain. For example,

it may be impossible to define the pl function by extension because the number of initial accessible situations may be infinite.

That is the case, for example, when the agent believes in the initial situation that the toy is not broken, and he ignores his position, and the number of possible position is infinite.

Fortunately, we can avoid having to give an extensional definition of pl by using the assumption: $Knows(\neg broken, s_0)$. Nevertheless, if we want to guarantee consistent beliefs after revision, we need to add the assumption: $broken(s_0) \rightarrow \exists s''(K(s'', s_0) \wedge \neg K_{max}(s'', s_0) \wedge broken(s''))$, where $K_{max}(s'', s_0)$ means that in the situation s_0 , the situation s'' is one of the most plausible.

If the toy is broken, is this assumption sufficient to prove that in $do(look, s_0)$ the agent believes that the toy is broken? The answer is not obvious.

This simple example shows that it is definitely not easy to guarantee that no assumption is missing in the description of the initial situation. If in the application domain there are five, ten, or even more fluents, modeling the initial situation may be quite problematic.

To remove these difficulties, in [2] Demolombe and Pozos Parra have proposed a simple but restrictive solution to belief revision. The idea is to generalize the successor state axioms from literals to modal literals. Although in this solution the scope of the beliefs is limited to literals, Petrick and Levesque in [5] have analyzed its expressive power, and they have shown that its limitations are not so strong. In [1] Demolombe and Herzog have also shown that a very similar idea has been formalized in the Dependence Logic.

2 Belief Revision Without Plausibility Levels

Here a new solution is proposed⁴ to belief revision. The most important feature is not to use plausibility levels and instead incorporate the sensing actions into the successor state axioms. Then the effect of a sensing⁵ action is **not to remove** the situations that are inconsistent with the perceived information, as Scherl and Levesque do in [8], but to change the truth values of the fluents in the imaginary successor situations according to the perceived information, in a similar way as non knowledge producing actions do.

There are two other significant features in this new proposal. The first one is that we may have **distinct successor state axioms** for real situations and for agents' imaginary situations. The second one is that we have an explicit representation of the actions whose performance can be observed by an agent, and, if an agent has not observed the performance of an action, then this action does not change his beliefs.

Before to present a general framework we analyze a particular scenario and its formalization.

⁴ A preliminary version of this work has been presented in [3].

⁵ In the following "sensing action" will be used as a shorthand for "knowledge producing action".

Scenario

In a room there is a young baby and his mother. It is assumed that the baby observes every action he performs iff he has his eyes open. The same applies for the mother.

In the situation s_0 (see Fig. 1)⁶ the baby holds a fragile toy in his hands. The toy is not broken and both the mother and the baby believe that the toy is not broken. The baby and the mother have their eyes open.

In the situation s_1 the baby has dropped the toy, i.e. $s_1 = do(drop(b), s_0)$ ⁷. The mother believes that a fragile toy will break after it is dropped. It follows that she believes that the toy is broken. She does not need to look at the toy to hold this belief. However, the baby does not believe that dropping a fragile toy will cause it to break. So the baby in s_1 believes that the toy is not broken.

In the situation s_2 the baby has looked at the toy, i.e. $s_2 = do(look(b), s_1)$, and he observed that it is broken. Then, in s_2 he believes that the toy is broken. In s_2 the mother still believes that the toy is broken.

In the situation s_3 the baby has closed his eyes, i.e. $s_3 = do(close(b), s_2)$. The baby's beliefs and the mother's beliefs **remain unchanged**, except about the fact that this action has been performed.

In the situation s_4 the mother has repaired the toy, i.e. $s_4 = do(repair(m), s_3)$. Since the baby has closed his eyes he ignores that this action has been performed. Then, in s_4 he still believes that the toy is broken.

The main features of this scenario are the following ones.

- In s_4 the baby ignores that the action $repair(m)$ has been performed. Then, in s_4 the baby holds the same beliefs than in s_3 , i.e. the baby's imaginary situations, like s'_3 , **remain unchanged** after the execution of the unobserved action.
- The mother and baby have different beliefs about the evolution of the fluent *broken*. That is why, even when both of them know initially the toy is not broken, we have $broken(s'_1)$ and $\neg broken(s'_1)$ (see Fig. 1).
- In s_2 the action $look(b)$ has informed the baby about the truth value of the fluent *broken* in s_1 . Though imaginary situations like s'_1 represent a belief that is false in s_1 , the situation s'_1 is not removed, and instead the baby changes in s'_2 the belief that the toy is not broken because we have $broken(s_1)$.

Formalization

The following notations are adopted.

$open(i)$: the agent i opens his eyes.

$close(i)$: the agent i closes his eyes.

$drop(i)$: the agent i drops the toy.

$look(i)$: the agent i looks whether or not the toy is broken.

⁶ To make the figure easier to read we have respectively used the notations K_m and K_b for $K(m, -, -)$ and $K(b, -, -)$.

⁷ In this section, since we may have several agents, action function symbols have an argument to make explicit the agent who is doing the action.

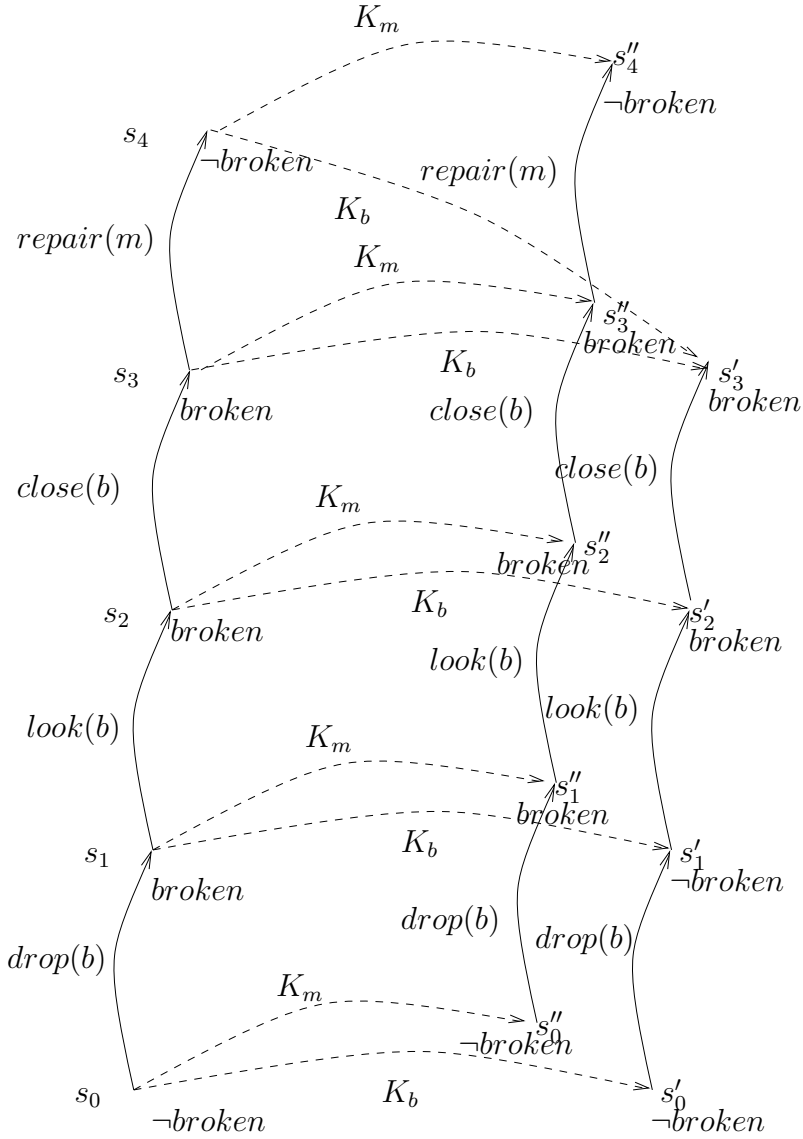


Fig. 1. Mother and baby beliefs' evolution

$repair(i)$: the agent i repairs the toy.

$broken(s)$: in the situation s the toy is broken.

$real(s)$: s is a real situation.

The person who is modeling an application defines which situations are real situations.

$observe(i, a, s)$: in the situation s , if the action a is performed, then after performance of a the agent i will be informed that a has been performed.

Note that $observe(i, a, s)$ does not mean that i has observed performance of a , but that i has the ability to observe performance of a **if** a is performed.

$K(i, s', s)$: in the situation s the agent i believes that s' is a situation where he might be in. That is, s' is compatible with what i believes in s .

$B(i, \phi(s', s), s', s)$: $\phi(s', s)$ holds in every situation s' related by K with i and s . In other words $\phi(s', s)$ is compatible with what i believes in s . Note that the free variables s' and s of ϕ must be the third and the fourth argument of B respectively, since s may not be a free variable of ϕ . In $B(i, \phi(s', s), s', s)$ the situation s may be a real situation or an imaginary situation.

Notice that B is an abbreviation and not a fluent. We have:

$$B(i, \phi(s', s), s', s) \stackrel{\text{def}}{=} \forall s'(K(i, s', s) \rightarrow \phi(s', s))$$

The formula $\phi(s', s)$ may be any formula in a first order language with the only restriction being that the fluent K only appears in formulas that represent beliefs.

Evolution of Imaginary Situations

If the action a is performed in the situation s , the situations that are accessible from $do(a, s)$ are the successors of the situations accessible from s if the action a has been observed by i , and they are the same situations as the situations accessible from s if the action a has not been observed by i . Then, we have:

$$(EK) \quad \forall s \forall s'' \forall a \forall i (K(i, s'', do(a, s)) \leftrightarrow \exists s' (K(i, s', s) \wedge ((observe(i, a, s) \wedge s'' = do(a, s')) \vee (\neg observe(i, a, s) \wedge s'' = s')))))$$

We can infer from (EK) that after an action has been performed an agent will believe it has been performed if he observes it.

We have to define a SSA for the fluent $observe(i, a, s)$. This SSA depends on each application domain. For the scenario we have presented here we have adopted the following SSA:

$$(OBS) \quad \forall s \forall a \forall a' \forall i (observe(i, a, do(a', s)) \leftrightarrow a' = open(i) \vee observe(i, a, s) \wedge \neg(a' = close(i)))$$

The intuitive meaning of (OBS) is that an agent has the capacity to observe any action provided he has opened his eyes and not subsequently closed them. Since s is not restricted to real situations, the SSA (OBS) is believed by every agent⁸.

The SSAs for the fluents depend on the context where we are. We may have different SSAs for the same fluent depending on the fact that we are in a real situation or in some agent's imaginary situation.

The SSA for the real situations is:

$$(SSA_r) \quad \forall s \forall a (real(s) \rightarrow (broken(do(a, s)) \leftrightarrow \exists i (a = drop(i)) \vee broken(s) \wedge \neg(\exists i (a = repair(i)))))$$

⁸ As a matter of simplification we have assumed that an agent performs no action, except open eyes if he has closed his eyes.

The SSA for the mother's imaginary situations is:

$$(SSA_m) \quad \forall s \forall s' \forall a (real(s) \rightarrow (K(m, s', s) \rightarrow broken(do(a, s')) \leftrightarrow \exists i (a = drop(i)) \vee (a = look(m) \wedge broken(s)) \vee broken(s') \wedge \neg(\exists i (a = repair(i)) \vee (a = look(m) \wedge \neg broken(s)))))$$

Notice that the fluent *broken* comes to be true (resp. false) in a mother's imaginary situation $do(a, s')$ if the mother has looked at the toy (action $look(m)$) and the toy is broken (resp. not broken) in the real situation s .

The SSA for the baby's imaginary situations is:

$$(SSA_b) \quad \forall s \forall s' \forall a (real(s) \rightarrow (K(b, s', s) \rightarrow broken(do(a, s')) \leftrightarrow (a = look(b) \wedge broken(s)) \vee broken(s') \wedge \neg(\exists i (a = repair(i)) \vee (a = look(b) \wedge \neg broken(s)))))$$

If we adopt the notations:

$$ssa(b, s', s) \stackrel{\text{def}}{=} broken(do(a, s')) \leftrightarrow (a = look(b) \wedge broken(s)) \vee broken(s') \wedge \neg(\exists i (a = repair(i)) \vee (a = look(b) \wedge \neg broken(s)))$$

$$ssa(m, s', s) \stackrel{\text{def}}{=} broken(do(a, s')) \leftrightarrow \exists i (a = drop(i)) \vee (a = look(m) \wedge broken(s)) \vee broken(s') \wedge \neg(\exists i (a = repair(i)) \vee (a = look(m) \wedge \neg broken(s)))$$

the axioms SSA_b and SSA_m can be reformulated as:

$$(SSA_b) \quad \forall s \forall a (real(s) \rightarrow B(b, ssa(b, s', s), s', s))$$

$$(SSA_m) \quad \forall s \forall a (real(s) \rightarrow B(m, ssa(m, s', s), s', s))$$

It is worth noting that, depending on the application, it can be assumed that an agent is informed about the SSAs believed by the other agents.

For example, it may be assumed that the baby believes that his mother believes, like him, that the toy is not broken if it is dropped, and also that the mother believes that her baby believes that the toy is not broken if it is dropped, even if she does not hold that belief.

That can be formally represented by the following axioms.

$$(SSA_{bm}) \quad \forall s \forall a (real(s) \rightarrow B(b, B(m, ssa(b, s'', s'), s'', s'), s', s))$$

$$(SSA_{mb}) \quad \forall s \forall a (real(s) \rightarrow B(m, B(b, ssa(b, s'', s'), s'', s'), s', s))$$

Successor State Axioms in general

In general, to define the “real” evolution of fluents, for each fluent p we have the SSA:

$$(SSA_p) \quad \forall s \forall a \forall \mathbf{x} (real(s) \rightarrow (p(\mathbf{x}, do(a, s)) \leftrightarrow \Gamma_p^+(\mathbf{x}, a, s) \vee p(\mathbf{x}, s) \wedge \neg \Gamma_p^-(\mathbf{x}, a, s)))$$

To define the “subjective” evolution of fluents, for an agent i and a fluent p we have the SSA:

$$(SSA_{i,p}) \quad \forall s \forall s' \forall a \forall \mathbf{x} (real(s) \rightarrow (K(i, s', s) \rightarrow (p(\mathbf{x}, do(a, s')) \leftrightarrow \Gamma_{i,p}^+(i, \mathbf{x}, a, s') \vee (a = sense_p(i) \wedge p(\mathbf{x}, s)) \vee p(\mathbf{x}, s') \wedge \neg(\Gamma_{i,p}^-(i, \mathbf{x}, a, s') \vee (a = sense_p(i) \wedge \neg p(\mathbf{x}, s)))))))$$

It is assumed that the successors of the real situations are real situations. Then, we have:

$$(SSA_R) \quad \forall s \forall a (real(do(a, s)) \leftrightarrow real(s))$$

Progression and Regression

For reasoning about actions and beliefs we define a Basic Action and Belief Theory T which is composed of the following axioms.

1. Σ : the foundational axioms for situations (see [7]).
2. T_{SS} : a set of successor state axioms of the form⁹ (SSA_p) and $(SSA_{i,p})$.
3. T_K : the successor state axiom (EK) for the fluent $K(i, s', s)$.
4. T_O : the successor state axiom for the fluent $observe(i, a, s)$ which depends on the application domain and has a form similar to (OBS) .
5. T_R : the successor state axiom (SSA_R) .
6. T_{AP} : a set of precondition axioms (see [7]).
7. T_{UNA} : a set of unique name axioms for actions (see [7]).
8. T_0 : a set of first order sentences defining the initial situation (including initial beliefs).

Progression

The progression problem is to infer from the initial situation s_0 consequences about properties in a further situation s_i . So, to solve the progression problem for $\phi(s_i)$, where s_i is a ground term, we have to prove that:

$$\vdash T \rightarrow \phi(s_i)$$

That can be proved with any standard automated theorem proving technique for classical first order logics with equality.

Regression

The regression problem is to find a transformation R_T that transforms a formula $\phi(do(a, s))$ about $do(a, s)$ into a formula $\psi(s)$ about s (i.e.

$R_T(\phi(do(a, s))) = \psi(s)$), and such that:

$$\vdash T \rightarrow (\phi(s_i) \leftrightarrow R_T^*(\phi(s_i)))$$

where s_i is a ground term and R_T^* represents an iterated application of R_T until the result of R_T application remains unchanged (in that case $R_T^*(\phi(s_i))$ is a sentence about s_0) (see [7]).

We say that R_T is **sound** if we have: $\vdash T \rightarrow R_T^*(\phi(s_i)) \Rightarrow \vdash T \rightarrow \phi(s_i)$.

We say that R_T is **complete** if we have: $\vdash T \rightarrow \phi(s_i) \Rightarrow \vdash T \rightarrow R_T^*(\phi(s_i))$.

At the present time we have not solved the regression problem in general. However, we can easily show that, if the SSAs are the same in every situations, a “rewriting” technique very close to the one defined in [8] can be proved to be sound and complete.

If the SSAs are different in imaginary situations, the problem is more complex and requires further investigations.

⁹ We omit for simplicity the axioms for functional fluents.

3 Conclusion

We have presented a general framework for belief revision in the Situation Calculus that does not require the definition of a plausibility distribution. The key idea is to have for each agent i SSAs for beliefs of the form of $SSA_{i,p}$.

These axioms define how the truth values of the fluents change in imaginary situations when a sensing action is performed, and we do not have to remove the imaginary situations that are inconsistent with the observations as Scherl and Levesque do in [8]. Also, it is worth noting that in the proposed framework iterated belief revisions do not raise specific problems.

The framework is dramatically simplified when the SSAs are independent of the context. In that case it is simpler to define regression.

In this approach the sensing actions provide information about the truth of a fluent and not about the truth of a general formula, like in [8]. Is this a strong limitation? We do not think so, because the information that is directly delivered by a sensor is always an atomic information, whose interpretation, in the theory of the application domain, may be represented by a general formula.

References

1. R. Demolombe and A. Herzig. Obligation change in Dependence Logic and Situation Calculus. In A. Lomuscio and D. Nute, editors, *Proceedings of the 7th International Workshop on Deontic Logic in Computer Science*. Springer, LNAI 3065, 2004.
2. R. Demolombe and M. P. Pozos-Parra. A simple and tractable extension of situation calculus to epistemic logic. In Z. W. Ras and S. Ohsuga, editors, *Proc. of 12th International Symposium ISMIS 2000*. Springer, LNAI 1932, 2000.
3. R. Demolombe and M. P. Pozos-Parra. Belief change in the situation calculus: a new proposal without plausibility levels. In A. Herzig and H. van Ditmarsch, editors, *Workshop on Belief Revision and Dynamic Logic, European Summer School on Logic, Language and Information*, 2005.
4. J. McCarthy. Situations, actions and causal laws. In M. Minski, editor, *Semantic Information Processing*. The MIT press, 1968.
5. R. Petrick and H. Levesque. Knowledge equivalence in combined action theories. In *Proceeding 8th International Conference on Principles of Knowledge Representation and Reasoning*, 2002.
6. R. Reiter. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 359–380. Academic Press, 1991.
7. R. Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.
8. R. Scherl and H. Levesque. The Frame Problem and Knowledge Producing Actions. In *Proc. of the National Conference of Artificial Intelligence*. AAAI Press, 1993.
9. S. Shapiro, M. Pagnuco, Y. Lespérance, and H. Levesque. Iterated belief change in the situation calculus. In *Proc. of the 7th Conference on Principles on Knowledge Representation and Reasoning (KR2000)*. Morgan Kaufman Publishers, 2000.

Norms, Institutional Power and Roles: Towards a Logical Framework

Robert Demolombe¹ and Vincent Louis²

¹ ONERA Toulouse
France

`Robert.Demolombe@cert.fr`

² France Telecom Research & Development
Lannion
France

`vincent.louis@francetelecom.com`

Abstract. In the design of the organisation of a multiagent system the concept of role is fundamental. We informally analyse this concept through examples. Then we propose a more formal definition that can be decomposed into: the conditions that have to be satisfied to hold a role, the norms and institutional powers that apply to a role holder. Finally, we present a modal logical framework to represent these concepts.

1 Introduction

In the design of the organisation of a multiagent system the concept of role is fundamental. That is especially important in the context of intelligent artificial agents that have some level of autonomy. However there is no agreement in the literature about what a role is [15,8,5]. The reason is that the concept of role refers to concepts like rights and powers, which are not very well defined. Therefore, there is a need for a formal definition of roles to make it possible to reason clearly about their properties.

In this paper¹ we start from an informal analysis of these concepts which is based on typical examples (section 2). From the synthesis of this analysis a more structured definition of roles is proposed, and then a possible way of formalising it in a modal logical framework is shown (section 3).

2 Informal Analysis of the Concept of Role

2.1 Definition

In natural language the term “role” may have many different meanings. For instance in some French dictionary we can find this definition in the context of sociology: *implicit or explicit set of rights and obligations about some individual*

¹ This research has been funded by France Telecom’s Research & Development division.

in a social group, in connection with her or his legal status or her or his position in this group. In the following, such “social groups” will be called “institutions”.

In [15], Pörn proposes this definition: *“Because of their prevalence in normative systems clusters of norms organised in this way deserve a name on their own. We shall call them role structure because in terms of them it is possible to define the sociological notion of a role.”* In other words, if, in a normative system, we repeatedly need to talk about the set of individuals to whom a given set of norms applies, it is convenient to create a name for this set of norms, and this set of norms is called a “role”. Later on, Pörn refines his definition, and he says that there are two components in a role:

1. a set of **conditions** that characterises the individuals who hold this role,
2. a set of **norms** that applies to the individuals who hold this role.

For instance, the role of minor is defined by a descriptive property: to be less than 18 years old, and by normative properties, for instance, the obligation to have the parents’ authorisation to go to a foreign country. In the same way, we can talk about the roles of supplier and customer in the trade environment, or to talk about the roles of professor, lecturer, PhD student or secretary, in the environment of a university.

We can give more details about what the conditions and the norms are in the case of the role of professor (in French universities):

1. Conditions: to be less than 65 years old, to have a PhD, to have been appointed to the position of professor, etc.
2. Norms: obligation to give x lectures a year, obligation to prepare the exams, prohibition to talk about politics during lectures, right to have an office, power to give marks to the students, etc.

It is worth noting that though the conditions in the part 1 are descriptive, it may happen that the truth value of some of them is defined only in the context of a particular institution. For example, the fact that some individual has a PhD may be recognised in some countries, and not recognised in other countries. In other words, the fact: “to have a PhD” depends on the referred institution. This kind of facts will be called in the following “**institutional facts**” (see [16]) and the term “**institution**” will refer to entities like: a state, a company, an association or any group organised for a common purpose.

Obviously, the norms in the part 2 are dependent on an institution, but the difference between **norms** and **institutional facts** is that the former ones define how the world **should** be, while the latter ones define how the world **is**.

To sum up, the conditions in a role definition specify descriptive properties, and some of them depend on an institution. The norms specify normative properties and all of them depend on an institution.

In the following we will use the term “agent” for an individual who may be a person or an artificial system that has some degree of autonomy.

2.2 From Normative Positions to Institutional Power

Now we analyse which kinds of norms can be found in the definition of a role.

In 1913, Hohfeld expressed in natural language the first attempt to classify the normative relationships between two agents [9]. Later, Kanger formalised all these normative positions using only the primitive concepts of obligation and action [12].

Back to roles, according to some dictionaries, a role can also be defined as a set of rights and obligations. As proposed by Kanger, we now investigate how to express the notion of right only in terms of obligations applying to actions.

Some possible dictionary definitions of right are: “*possibility to require something from someone*”, or: “*something that is due to someone in a social group*”. In both of these definitions there is the idea that an agent who has some right can require something of another agent, and this capacity depends on an institution. This concept of right can be made clearer through the following example.

At ONERA a person who holds the role of engineer has the right to enter the centre with his car. To exercise his right ² when he arrives at the entrance with his car, the engineer must show to the guard a sign on the glass of the car, which testifies that he has the right to enter.

When the guard sees the sign, he knows that the engineer has the right to enter. Then, if the entrance gate is not raised, the guard has the obligation to raise the gate, and if the entrance gate is raised, the guard is prohibited to prevent the engineer to enter (for example, by lowering the gate).

From this example we can extract a more general pattern which is based on the notion of power. An agent i who has some right is an agent who has the power, when he requires another agent j to exercise his right, and when he has testified that he has this right, to create new obligations that apply to j . The difference between Hohfeld’s definition of a right and this definition is that in this definition we point out the power to change the obligations.

The kind of power we have exhibited in this example belongs to what Pörn calls the norms in his role definition. To avoid misunderstanding between this kind of power and practical ability we will use in the following the term: “institutional power” as in [10].

Even if the concept of right can be defined in terms of institutional power it is not the case that any institutional power is of the type of a right, as shown through the following example of the power to appoint a professor.

A person who holds the role of president of the university has the power to appoint a professor. That means that after some given selection procedure has been performed, if he signs some particular document, then the person who applied to the role of professor holds the role of professor.

The relationship between the “cause” (to sign the document) and the “consequence” (to hold the role of professor) has a value only in the context of the institution of the university. It is worth noting that here the consequence is not an obligation, as it is the case for a right, but an institutional fact.

² We have slightly simplified the actual procedure.

This example shows that if, in the context of a given institution, an agent has some institutional power, then, if in some circumstances he performs some procedure, he creates a new normative situation.

In general an institutional power is defined as follows:

$power(i, s, cond, proc, n)$: if the conditions $cond$ are fulfilled, then, in the context of the institution s , the agent i has the power to create the institutional fact represented by n by performing the procedure $proc$.

In semi-formal terms we have:

IF $cond$ AND $done(i, proc)$ THEN n [in the context of s].

The main difference between the institutional powers and the normative positions defined by Hohfeld is that the institutional powers allow to change a normative situation, while the normative positions define the normative situation as it is at a given moment.

In [14] Makinson has shown that we can define an extremely large number of normative positions. Then, the Hohfeld's classification is not based on the most relevant criteria. For that reason we have only distinguished two kinds of norms: norms like normative positions, which will be called in the following **static** norms, and institutional powers, which will be called **dynamic** norms.³

2.3 A New Definition of Roles

We propose to define a role in the context of an institution s as a tuple $\langle R, C, N, P \rangle$, such that:

1. R is the role name.
2. C is a set of descriptive sentences that defines the necessary and sufficient conditions that are satisfied by an agent who holds the role R .
3. N is the set of **all** the static norms that apply to an agent as holder of the role R .
4. P is the set of **all** the dynamic norms (institutional powers) that apply to an agent as holder of the role R .

This definition is basically a refinement of Pörn's definition. It requires some additional comments.

- A role is always defined in the context of an institution. For instance, the role of president is not the same in all countries. As a matter of simplification the name of the institution is left implicit if there is no risk of confusion.
- The sentences in C are descriptive sentences. They describe the properties that **are** satisfied by the role holder, and not the properties that **should be** satisfied.
- The sets of norms N and P apply to any agent who holds the role R . However, it may happen that other norms apply to such an agent. For instance, because he also holds another role.

³ Not all the institutional powers represent dynamic norms. There are institutional powers that allow to create descriptive institutional facts like: to hold a role (see the formal definition proposed in next sections).

- The norms in N may be conditional norms. In particular we frequently distinguish the norms that apply only when the agent exercises his role. To do so, it is more convenient to split N in two subsets: the norms N_H that apply in every circumstances, and the norms N_E that apply only when the agent exercises his role. In this case, the set C_E of necessary and sufficient conditions that are satisfied by an agent exercising his role has to be defined, in addition to C .

For example, it is forbidden for a professor to talk about politics when he is exercising his role at the university, but he is authorized to talk about politics outside the university.

- It is assumed that all the dynamic norms in P are institutional powers of the form $power(i, s, cond, proc, n)$.

Here $power$ is not a predicate but a notation to denote a sentence of the form: IF $cond$ AND $done(i, proc)$ THEN n [in the context of s]

where $proc$ is a procedure definition, and $cond$ is a conjunction of descriptive sentences.

- In the definition of an institutional power, the institutional fact n may be of different kinds. For instance, it may be the fact that an obligation holds, the fact that an agent j is appointed to another role R' , or the fact that j has a certain institutional power.

A First Step Towards Formalisation. We define the following predicates.

$holds(i, R)$: agent i holds the role R .

$exercises(i, R)$: agent i exercises the role R .

$done(i, proc)$: agent i has performed the procedure $proc$.

It is assumed that the following properties are satisfied by a given role R .

(H1) $\forall i$ (IF $holds(i, R)$ THEN $C(i, R)$)

(H2) $\forall i$ (IF $C(i, R)$ THEN $holds(i, R)$)

(N1) $\forall i$ (IF $holds(i, R)$ THEN $N(i, R)$)

(P1) $\forall i$ (IF $holds(i, R)$ THEN $P(i, R)$)

In the set of conditions to hold a role, $C(i, R)$, the fact that the agent i has been appointed to the role R by some other agent j can be formally represented, for example, by statements of the form $\exists j done(j, proc)$.

In order to make explicit the norms that only apply when an agent exercises his role, (N1) has to be replaced with the following properties:

(E1) $\forall i$ (IF $exercises(i, R)$ THEN $C_E(i, R)$)

(E2) $\forall i$ (IF $C_E(i, R)$ THEN $exercises(i, R)$)

(N'1) $\forall i$ (IF $holds(i, R)$ THEN $N_H(i, R)$)

(N''1) $\forall i$ (IF $holds(i, R)$ AND $exercises(i, R)$ THEN $N_E(i, R)$)

In the set of dynamic norms, $P(i, R)$, the institutional powers the agent i has only in circumstances where he exercises his role have to be defined by specifying appropriately (using the $exercises(i, R)$ predicate) the conditions $cond$ of the definition of the corresponding powers: $power(i, s, cond, proc, n)$.

3 Formalisation of the Concept of Role

We are now defining the main properties of a logical framework that allows to represent the most significant concepts that are involved into role definitions.

In the sentences that appear in C , N , $cond$ and n there may be sentences that are not in the scope of any modality. These sentences can be represented in a classical first order logic. There may also be sentences about agents' beliefs and actions that have been performed. These sentences respectively require the definition of an epistemic logic and of a dynamic logic. Finally, sentences about normative situations require the definition of a deontic logic.

Our guideline in the definition of this logical framework is to adopt an axiomatics as simple as possible for these logics.

3.1 Epistemic and Dynamic Modalities

Beliefs are represented by the modalities:

$B_i p$: agent i believes p .

Where p may be any sentence, and involve other modalities. We adopt the system (KD) for the modality B_i (see [4]).

To represent dynamic modalities we adopt two modal operators. The first operator is used when we only need to represent the effects of the action that has been performed without making explicit this action (for example, to represent the fact that it is obligatory for the guard to bring it about that the entrance gate is raised). The second operator is used to explicitly represent the action, or the procedure, that has been performed (for example, to represent the procedure $proc$ in an institutional power).

$E_i p$: agent i has brought it about that p .

$done(i, proc, p)$: agent i has just performed the procedure $proc$, and before $proc$ was performed, we had p .

$done(i, proc)$: abbreviation for $done(i, proc, true)$, whose meaning is that agent i has just performed $proc$.

The modal operator E_i is a classical operator which is not normal (see [4]), its axiomatics is (see [15,11]):

(RE) $\vdash p \leftrightarrow q \Rightarrow \vdash E_i p \leftrightarrow E_i q$

(C) $E_i p \wedge E_i q \rightarrow E_i(p \wedge q)$

($\neg N$) $\neg E_i(true)$

(T) $E_i p \rightarrow p$

The modal operator $done$ is a normal operator which satisfies the system (K). This operator is basically used to represent the procedure that has to be performed by an agent to exercise his power.

We have not enough room here to go deeply into the language which is used to represent the procedure $proc$. The following example: $proc = A; (any/B); (C|D)$, shows the main kinds of constructs that can be used to represent procedures that may be performed either by human agents or by artificial agents. Its intuitive meaning is: do A , then it is permitted to do any sequence of actions but B , and then do C or D . The formal semantics of this language can be found in [7].

3.2 Deontic Modalities

Deontic modalities play a fundamental role for the representation of normative sentences. They express what is obligatory, permitted or prohibited. A large number of proposals exist in the literature to represent these modalities (see [15,3,13,19,1]), and depending on the kind of issue under consideration, some axiomatics or another can be chosen.

However, we need to consider at least two kinds of deontic modalities to represent the obligations to be (for instance, the obligation to be sitting during a lecture), and the obligations to do (for instance, the obligation to bring some object somewhere).

Some authors also consider personal obligations (for instance, the obligation for a given agent to pay a given bill) and impersonal obligations that do not refer to a particular agent (for instance, the obligation not to park a car in a given place). Here we only consider impersonal obligations. This is not too strong a limitation because personal obligations can be reformulated in terms of impersonal obligations about sentences where the obliged agent is explicitly mentioned.

Impersonal obligations to be are represented by the modality:

Op : it is obligatory that p .

Permissions and prohibitions can be respectively represented as usual in terms of obligations by: $\neg Op$ and $O\neg p$. The axiomatics of the operator O satisfies the system (KD).

Obligations to do require specifying the deadline before which the action must be performed. If no deadline is specified it is impossible to say when the obligation has been violated. There is a very limited number of proposals for a logic of obligations with deadline (see [18,17,2]). In [6] we have defined a logic for obligations, permissions and prohibitions with deadlines that extends the logic proposed by Segerberg in [17,18]. Finally, to simplify, we accept to represent conditional obligations using the material implication. The fact that p is obligatory in the circumstances where we have q is represented by: $q \rightarrow Op$. Contrary-to-duty obligations can be represented with the logic proposed in [3].

3.3 Normative Consequences

We have seen that in the semi-formal definition of institutional powers we have sentences of the form: "IF p THEN q [in the context of s]". We need a logical connective that correctly represents the link between p and q in these kinds of sentences.

We cannot use the material implication for that because $\neg p$ entails $p \rightarrow q$, and it would not be correct to infer that some agent has some institutional power from the fact that the conditions in the antecedent of the definition of this power are false. More specifically, if $power(i, s, cond, proc, n)$ is formally represented by: $cond \wedge done(i, proc) \rightarrow n$, from the fact that $\neg cond$ or $\neg done(i, proc)$, we can infer that the agent i has any power of the form $power(i, s, cond, proc, n)$, whatever n is.

That is the reason why we adopt the following connective to represent normative consequences, which has been defined by Jones and Sergot in [10]:

$p \Rightarrow_s q$: in the context of the institution s , p entails q .

In addition to the connective $\Rightarrow_s$ Jones and Sergot have introduced the modal operator:

$D_s p$: in the context of the institution s we have p .

For instance, if the meaning of p is that two persons are married and s represents a given state, $D_s p$ means that, in the context of the regulation of this state, these persons are considered as two legally married persons. However, it may happen that with respect to the regulation of another state s' they are not married.

The D_s modality satisfies the system (KD). The axiomatics of $\Rightarrow_s$ is defined as follows.

For the antecedent and the consequent we have the inference rules of substitutivity of logically equivalent formulas. In addition we have the axiom schemas:

$$(CC) \quad (p \Rightarrow_s q) \wedge (p \Rightarrow_s q') \rightarrow (p \Rightarrow_s (q \wedge q'))$$

$$(CA) \quad (p \Rightarrow_s q) \wedge (p' \Rightarrow_s q) \rightarrow ((p \vee p') \Rightarrow_s q)$$

$$(S) \quad (p \Rightarrow_s q) \rightarrow ((q \Rightarrow_s r) \rightarrow (p \Rightarrow_s r))$$

The links between $\Rightarrow_s$ and D_s are expressed by the axiom schemas:

$$(SD) \quad (p \Rightarrow_s q) \rightarrow D_s(p \rightarrow q)$$

$$(SC) \quad (p \Rightarrow_s q) \rightarrow (p \rightarrow D_s p)$$

We also adopt the following additional schemas⁴:

$$(DD) \quad D_s D_s p \rightarrow D_s p$$

$$(DP) \quad D_s(p \Rightarrow_s q) \rightarrow p \Rightarrow_s q$$

3.4 Formal Representation of Roles

The role definitions are formalised using $\Rightarrow_s$ and D_s . We have:

$$(H1) \quad \forall i D_s(\text{holds}(i, R) \rightarrow C(i, R))$$

$$(H2) \quad \forall i (C(i, R) \Rightarrow_s \text{holds}(i, R))$$

$$(N1) \quad \forall i D_s(\text{holds}(i, R) \rightarrow N(i, R))$$

$$(P1) \quad \forall i D_s(\text{holds}(i, R) \rightarrow P(i, R))$$

Note that the sentences of the form: IF...THEN... are not all formalised in the same way. This depends on whether IF...THEN... represents an entailment relation or if the antecedent counts as the consequent in the context of the institution s . In particular, in the context of s , $C(i, R)$, $N(i, R)$ and $P(i, R)$ are necessarily true when $\text{holds}(i, R)$ is true.

For example, the fact that if i holds the role of minor then i is less than 18 years old does not mean that the fact that i holds the role of minor counts as the fact that i is less than 18 years old. That is why in (H1) we have used a material implication in the scope of D_s . In (N1) and (P1) we have used the material implication for the same reason.

In (H2) the fact that we have $C(i, R)$ counts as the fact that we have $\text{holds}(i, R)$ in the context of s . That is why we have used the connective $\Rightarrow_s$.

⁴ The schemas (DD) and (DP) are not in [10].

An institutional power denoted by $power(i, s, cond, proc, n)$, is formally represented by:

$$(cond \wedge done(i, proc)) \Rightarrow_s n$$

Let us illustrate this logical framework with a simple example. Let us assume that: (1) when an agent i holds the role R_1 , he has the institutional power to appoint j to the role R_2 , (2) i has been appointed to the role R_1 , and (3) i has performed the appropriated procedure to appoint j to R_2 in circumstances where the conditions $cond$ hold. These assumptions are formally represented by:

$$(1) \quad D_s(holds(i, R_1) \rightarrow power(i, s, cond, proc, holds(j, R_2)))$$

$$(2) \quad D_s holds(i, R_1)$$

$$(3) \quad cond \wedge done(i, proc)$$

From (1) and (2) we have: (4) $D_s(power(i, s, cond, proc, holds(j, R_2)))$, and from the definition of $power$ we have:

$$(5) \quad D_s(cond \wedge done(i, proc) \Rightarrow_s holds(j, R_2))$$

From (5) and (DP) we have:

$$(6) \quad cond \wedge done(i, proc) \Rightarrow_s holds(j, R_2)$$

From (6) and (SD) we have:

$$(7) \quad D_s((cond \wedge done(i, proc)) \rightarrow holds(j, R_2))$$

From (6) and (SC) we have:

$$(8) \quad (cond \wedge done(i, proc)) \rightarrow D_s(cond \wedge done(i, proc))$$

Finally, from (3), (8) and (7) we infer $D_s(holds(j, R_2))$, which means that in the context of s it is recognised that j holds the role R_2 .

It is interesting to notice that in $p \Rightarrow_s q$, the sentence q may itself embeb the connective $\Rightarrow_s$. This makes it possible to represent, for example, that the agent i has the power to assign to the agent j the power to create some normative situation n . This is denoted by:

$$power(i, s, cond, proc, power(j, s, cond', proc', n))$$

and this is formally represented by:

$$(cond \wedge done(i, proc)) \Rightarrow_s ((cond' \wedge done(j, proc')) \Rightarrow_s n)$$

4 Conclusion

We have shown that a role can be defined, in the context of an institution in terms of the conditions to hold the role, a set of static norms that apply to the agents who hold the role, and a set of dynamic norms that apply to the agents who hold the role.

We have presented a formal logical framework to represent the conditions and the norms. In this framework we have selected an axiomatics for epistemic, dynamic and deontic modalities. We have seen that the connective $\Rightarrow_s$ introduced by Jones and Sergot is very important to represent dynamic norms.

There are some technical issues that require more investigations. The first one is the formal representation of the fact that N and P represent **all** the static and dynamic norms that apply to a role holder. Another one is to find a complete axiomatics for the obligations to do with time limits.

References

1. L. Aqvist. Deontic logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, volume 2. Reidel, 1984.
2. J. Broersen, F. Dignum, V. Dignum, and J.-J. C. Meyer. Designing a deontic logic of deadlines. In A. Lomuscio and D. Nute, editors, *Proceedings of the 7th International Workshop on Deontic Logic in Computer Science*. Springer, LNAI 3065, 2004.
3. J. Carmo and A.J.I. Jones. Deontic Logic and Contrary-to-Duties. In D. Gabbay, editor, *Handbook of Philosophical Logic (Rev. Edition)*. Reidel, to appear.
4. B. F. Chellas. *Modal Logic: An introduction*. Cambridge University Press, 1988.
5. M. Dastani, V. Dignum, and F. Dignum. Role-assignment in open agent societies. In *Proceedings of the Second International Conference on Autonomous Agents and Multiagent Systems*, 2003.
6. R. Demolombe, P. Bretier, and V. Louis. Formalisation de l'obligation de faire avec délais. In *Troisièmes Journées francophones Modèles Formels de l'Interaction*, 2005.
7. R. Demolombe and E. Hamon. What does it mean that an agent is performing a typical procedure? A formal definition in the Situation Calculus. In C. Castelfranci and W. Lewis Johnson, editor, *First International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM Press, 2002.
8. M. Fasli. On commitments, roles and obligations. In *Second International Workshop of Central Eastern Europe on Multi-Agent Systems on: From theory to practice in multi-agent systems*, 2001.
9. W. N. Hohfeld. Some fundamental legal conceptions as applied in judicial reasoning. *Yale Law Journal*, 23, 1913.
10. A. J. Jones and M. Sergot. A formal characterisation of institutionalised power. *Journal of the Interest Group in Pure and Applied Logics*, 4(3), 1996.
11. A. J. I. Jones. A logical framework. In J. Pitt, editor, *The Open Agent Society*. John Wiley and Sons, To appear.
12. S. Kanger. New foundations of ethical theory. In R. Hilpinen, editor, *Deontic logic*, pages 36–58. D. Reidel Publishing Company, 1983.
13. C. Krogh. Obligations in multiagents systems. In *Proceedings of the fifth Scandinavian Conference on AI*. IOS Press, 1995.
14. D. Makinson. On the formal representation of rights relations. *Journal of Philosophical Logic*, 15, 1986.
15. I. Pörn. Action Theory and Social Science. Some Formal Models. *Synthese Library*, 120, 1977.
16. J. R. Searle. *Speech Acts: An essay in the philosophy of language*. Cambridge University Press, New-York, 1969.
17. K. Segerberg. Some Meinong/Chisholm thesis. In K. Segerberg and K. Sliwinski, editors, *Logic, Law, Morality. A festschrift in honor of Lennart Aqvist*, volume 51, pages 67–77. Uppsala Philosophical Studies, 2003.
18. K. Segerberg. Intension, Intention. In R. Kahle, editor, *To be announced*. CSLI Publications, 2004.
19. G. H. von Wright. *Norm and Action*. Routledge and Kegan, 1963.

Adding Efficient Data Management to Logic Programming Systems*

G. Terracina, N. Leone, V. Lio, and C. Panetta

Dipartimento di Matematica, Università della Calabria
Via Pietro Bucci, 87036 Rende (CS), Italy
{terracina, leone, lio, panetta}@mat.unical.it

Abstract. This paper considers the problem of reasoning on massive amounts of (possibly distributed) data. Presently, existing proposals show some limitations: *(i)* the quantity of data that can be handled contemporarily is limited, due to the fact that reasoning is generally carried out in main memory; *(ii)* the interaction with external (and independent) DBMSs is not trivial and, in several cases, not allowed at all; *(iii)* the efficiency of present implementations is still not sufficient for their utilization in complex reasoning tasks involving massive amounts of data. This paper provides a contribution in this setting; it presents a new system, called DLV^{DB} , which aims to solve all these problems.

1 Introduction

The problem of handling massive amounts of data received much attention in the research related to the development of efficient Database Management Systems (DBMSs). In this scenario, a mounting wave of data intensive and knowledge based applications, such as Data Mining, Data Warehousing and Online Analytical Processing have created a strong demand for more powerful database languages and systems. This led to the definition of several data model extensions, new language constructs, and various database extenders, to enhance the current capabilities and expressiveness of DBMSs. A great effort in this direction has been carried out with the introduction of a new standard for SQL, namely SQL99 [1] which provides, among other features, support to object oriented databases and recursive queries.

However, the adoption of SQL99 is still far from being a “standard”; in fact almost all current DBMSs do not fully support SQL99 and, in some cases, they adopt proprietary (non standard) language constructs and functions to implement parts of it. Moreover, the efficiency of current implementations of SQL99 constructs and their expressiveness are still not sufficient for performing complex reasoning tasks on huge amounts of data.

Complex reasoning capabilities can be provided by logic-based systems. In particular, Deductive Database Systems (DDSs) are advanced forms of database management systems, whose query languages, based on logics, are very

* This work has been partially supported by the italian “Ministero delle Attività Produttive” under project “Discovery Farm” B01/0297/P 42749-13, and by M.I.U.R. under project “ONTO-DLV: Un ambiente basato sulla Programmazione Logica Disgiuntiva per il trattamento di Ontologie” 2521.

expressive. DDSs not only store explicit information in the manner of a relational database, but they also store rules that enable deductive inferences based on the stored data. Using techniques developed for relational systems in conjunction with declarative logic programming, deductive databases are capable of performing reasoning based on that information.

However, current DDSs perform their reasoning tasks in main memory. This significantly limits the amount of data they can handle contemporarily, and does not fit with typical database applications requirements, where the huge amount of data implies the usage of secondary storage. Efficient access to this data is, then, crucial to achieve good performance. Unfortunately, the interaction of current DDSs with external (and independent) DBMSs is not trivial and, in several cases, not allowed at all.

Summarizing both DBMSs and DDSs presently show severe limitations for reasoning on massive amounts of data.

This work provides a contribution in this setting, bridging the gap between logic-based DDSs and DBMSs. It presents a new system, named DLV^{DB} , which has all the features of a DDS but can do all the work on mass memory and, in practice, does not have any limitation in the dimension of input data; moreover, it is capable to exploit optimization techniques both from DBMS (e.g., join orderings [8]) and DDS theory (e.g., magic sets [4,11]).

DLV^{DB} allows for two typologies of execution: (i) *Direct database* execution, which evaluates datalog programs directly on the databases through the execution of a suitable query plan of SQL statements, with a very limited usage of main memory; it is capable of handling massive amounts of data but with some limitations on the expressiveness of the queries (normal stratified datalog programs with aggregates are supported). (ii) *Main memory* execution, which loads input data from different (possibly distributed) databases and executes the datalog program directly into the main memory; in this case disjunctive datalog is fully supported.

In both cases, interoperability with databases is provided by ODBC connections; these allow to handle, in a quite simple way, data residing on various databases over the network.

There are several data intensive applications that could benefit from the adoption of DLV^{DB} . These are problems that cannot be efficiently managed neither by main memory logic systems nor by traditional databases, such as the automatic correction of census data [7]; the integration of inconsistent and incomplete data [10]; the elaboration on Scientific Databases in order to detect molecular features (e.g., gene functions).

Summarizing, the overall contributions of this work are the following: (i) the development of a fully fledged system enhancing in different ways the interactions between logic-based systems and DBMSs; (ii) the implementation of a mass-memory-based evaluation strategy for datalog programs allowing to minimize the usage of main-memory and to maximize the advantages of optimization techniques implemented in existing DBMSs; (iii) the execution of a thorough experimental analysis comparing the performance of state-of-the-art systems (DB2,

SQLServer, LDL++, XSB, and Smodels) with DLV^{DB} on classical DDS problems¹ [3]. The importance of our work, especially the direct database modality, lies both in the significantly larger amount of data that can be handled and in the increased efficiency in answering classical DDS queries (sometimes in orders of magnitude w.r.t. existing systems) and in its capability to take full advantage of optimization strategies from both logic and DBMS theory.

2 System Description

DLV^{DB} has been designed as an extension of the DLV system [5], both to handle input and output data distributed on several databases, and to allow the evaluation of datalog programs directly on databases. It combines the expressive power of DLV with the efficient data management features of DBMSs [8]. The language of DLV^{DB} is based on disjunctive datalog. We assume that the reader is familiar with the standard notions of logic programming and datalog [12].

As pointed out in the Introduction, DLV^{DB} allows for two typologies of execution: (i) main memory execution and (ii) direct database execution.

2.1 Main-Memory Execution

The main-memory execution of DLV^{DB} instantiates the input datalog program in main-memory (and, consequently, it exploits the standard DLV instantiator module), but allows input facts to be (possibly complex) views on database tables, which are stored in different DBMSs; moreover, it allows to export the result of the execution to database relations in DBMSs. These functionalities are not new from a scientific point of view but they represent a technological advancement w.r.t. analogous systems. This typology of execution supports the full DLV language and all its extensions and optimizations (like strong and weak constraints, external built-ins, magic-set optimizations, etc.).

In order to perform these tasks, two built-in commands are introduced in the DLV syntax, namely the `#import` and the `#export` commands:

```
#import(databasename,username,password,"query",predname, "typeConv").
#export(databasename,username,password,predname,tablename).
```

An `#import` command retrieves data from a table “row by row” through the *query* specified by the user in SQL and creates one atom for each selected tuple. The name of each imported atom is set to *predname*, and is considered as a fact of the program.

The `#export` command generates a new tuple into *tablename* for each new truth value derived for *predname* by the program evaluation.

Example 1. Assume that a travel agency asks to derive all the destinations reachable by an airline company either by using its vectors or by exploiting code-share agreements. Suppose that the direct flights of each company are stored in

¹ Note that other important DBMSs, such as Oracle, Postgres, and MySQL could not be tested. In fact, Oracle does not support SQL99, but only a simplified version of recursion, whereas Postgres and MySQL do not support recursive queries at all.

a relation `flight_rel(Id, From, To, Company)` of the database `dbAirports`, whereas the code-share agreements between companies are stored in a relation `codeshare_rel(Company1, Company2, FlightId)` of an external database `dbCommercial`; if a code-share agreement holds between the company `c1` and the company `c2` for *FlightId*, it means that the flight *FlightId* is actually provided by a vector of `c1` but can be considered also carried out by `c2`. Finally, assume that, for security reasons, it is not allowed to travel agencies to directly access the databases `dbAirports` and `dbCommercial`, and, consequently, it is necessary to store the output result in a relation `composedCompanyRoutes` of a separate database `dbTravelAgency` supposed to support travel agencies. The datalog program that can derive all the connections is:

- (1) $destinations(From, To, Comp) \text{ :- } flight(Id, From, To, Comp).$
- (2) $destinations(From, To, Comp) \text{ :- } flight(Id, From, To, C2), codeshare(C2, Comp, Id).$
- (3) $destinations(From, To, Comp) \text{ :- } destinations(From, T2, Comp),$
 $destinations(T2, To, Comp).$

In order to exploit data residing in the databases cited above, we should map the predicate *flight* with the relation `flight_rel` of `dbAirports` and the predicate *codeshare* with the relation `codeshare_rel` of `dbCommercial`. Finally, we have to map the predicate *destinations* with the relation `composedCompanyRoutes` of `dbTravelAgency`.

The commands that must be added to the datalog program above to implement these mappings are:

```
#import(dbAirports,airportUser,airportPasswd , "SELECT * FROM flight_rel" , flight,
                                     "type : U_INT, Q_CONST, Q_CONST, Q_CONST").
#import(dbCommercial,commUser,commPasswd , "SELECT * FROM codeshare_rel" , codeshare,
                                     "type : Q_CONST, Q_CONST, U_INT").
#export(dbTravelAgency,agencyName,agencyPasswd, destinations, composedCompanyRoutes).
```

2.2 Direct Database Execution

Language and functionalities. The direct database execution is the main innovation of our system; it supports normal stratified programs [12] and both true negation and negation as failure are supported in input programs. Moreover, all built-in predicates as well as aggregate functions [6] allowed in DLV are also supported by DLV^{DB} .

Three main peculiarities characterize DLV^{DB} in this execution modality: (i) its ability to evaluate datalog programs directly on databases with a very limited usage of main memory resources, (ii) its capability to map program predicates to (possibly complex and distributed) database views, and (iii) the possibility to easily specify which data is to be considered as input or as output for the program. As a consequence, it allows to easily integrate massive amounts of data residing in different databases and to perform complex deduction tasks on them.

Auxiliary directives. In order to allow the interaction with external (and independent) DBMSs, each datalog program can be associated with a set of directives specifying the needed mappings between database relations and predicates. The grammar in which these directives must be expressed is shown in Figure 1.

The user must specify the working database in which the system has to perform the instantiation. Moreover, he can specify a set of table definitions; note

```

Auxiliary-Directives ::= Init-section [Table-definition]+ [Final-section]*
Init-Section ::= USEDB DatabaseName:UserName:Password [System-Like]?.
Table-definition ::=
  [USE TableName [( AttrName [, AttrName]* )]? [FROM DatabaseName:UserName:Password]?
  [MAPTO PredName [( SqlType [, SqlType]* )]? ]??.
  |
  CREATE TableName [( AttrName [, AttrName]* )]?
  [MAPTO PredName [( SqlType [, SqlType]* )]? ]? [KEEP_AFTER_EXECUTION]?.]
Final-section ::= [DBOUTPUT DatabaseName:UserName:Password.] |
  OUTPUT [APPEND | REWRITE]? PredName [AS AliasName]? IN DatabaseName:UserName:Password.]

```

Fig. 1. Grammar of the auxiliary directives

that each specified table is mapped into one of the program predicates. Facts can reside on separate databases or they can be obtained as views on different tables. Attribute type declaration is needed only for a correct management of built-in predicates. The **USE** and **CREATE** options can be exploited to specify input and output data as well as temporary relations needed for the mass memory instantiation. Finally, the user can choose to copy the entire output of the instantiation or parts thereof in different databases.

Example 2. Consider again the scenario introduced in Example 1, and suppose that, due to a huge size of input data, we want to perform the execution in mass-memory (on a DBMS). In order to carry out this task, the auxiliary directives shown in Figure 2 should be used. They allow to specify the database relations that must be manipulated by the system.

Implementation issues. An input program $\mathcal{P}$ is first analyzed by the Parser which encodes the rules in a suitable internal format. After this, the Optimizer applies a rewriting procedure in order to get a program $\mathcal{P}'$, equivalent to $\mathcal{P}$, that can be instantiated more efficiently and that can lead to a smaller ground program. The Dependency Graph Builder computes the dependency graph of $\mathcal{P}'$, its connected components and a topological ordering of these components. Finally, the DB Instantiator module, the core of the system, is activated.

The DB Instantiator evaluates the input program through a bottom-up fix-point evaluation strategy which is based on the translation of each datalog rule (either recursive or not) in a corresponding non recursive SQL statement; this allows DLV^{DB} to be completely independent of the dimension of both the input data and the produced ground program. Since the input program is supposed to be normal and stratified, the DB Instantiator evaluates completely the program and no further modules must be employed after its execution.

The program evaluation strategy of DLV^{DB} corresponds to formulating a query plan, which follows the differential Semi-Naive method presented in [2].

```

USEDB dlvdb:myname:mypasswd.
USE flight_rel (Id, From, To, Company) FROM dbAirports:airportUser:airportPasswd
MAPTO flight (integer, varchar(255), varchar(255), varchar(255)).
USE codeshare_rel (Company1, Company2, FlightId) FROM dbCommercial:commUser:commPasswd
MAPTO codeshare (varchar(255), varchar(255), integer).
CREATE destinations_rel (From, To, Company)
MAPTO destinations (varchar(255), varchar(255), varchar(255)) KEEP_AFTER_EXECUTION.
OUTPUT destinations AS composedCompanyRoutes IN dbTravelAgency:agencyName:agencyPasswd.

```

Fig. 2. Auxiliary directives for Example 2

Specifically, a program $\mathcal{P}$ is evaluated one component at a time, starting from the lowest ones in the topological order according to the dependency graph associated with $\mathcal{P}$. The evaluation of each component is iterated until no new ground instance can be derived from it and the least fix-point is reached for $\mathcal{P}$.

Our differential implementation of the Semi-Naive method allows to significantly reduce the amount and the complexity of necessary joins to be carried out by: (i) Considering at the generic iteration k three relations for each predicate r , namely R^{k-2} storing the truth values computed for r until iteration $k-2$, ΔR^{k-1} storing only the new values computed at iteration $k-1$, and ΔR^k which stores only the new values computed for r at the current iteration. (ii) Translating each recursive rule in a set of non recursive rules each containing a suitable combination of standard and incremental relations. This allows to have smaller relations involved in the joins and a significantly smaller number of joins to be carried out during each iteration.

At each iteration of our algorithm, the instantiation of a rule consists of the execution of the SQL statement it is associated with. One of the main objectives in the implementation of DLV^{DB} has been that of associating one single (non recursive) SQL statement with each rule of the program (either recursive or not), without the support of main-memory data structures for the instantiation. This allows DLV^{DB} to fully exploit optimization mechanisms implemented in the DBMSs and to minimize the “out of memory” problems caused by limited main-memory dimensions.

Example 3. Consider the datalog program presented in Example 1 and the mappings shown in Figure 2. The query plan derived by DLV^{DB} for them is:

- (1) INSERT INTO destinations_rel
(SELECT f.From, f.To, f.Company FROM flight_rel AS f)
- (2) INSERT INTO destinations_rel
(SELECT f.From, f.To, c.Company2 FROM flight_rel AS f, codeshare_rel AS c
WHERE (f.Id=c.FlightId) AND (f.Company=c.Company1)
EXCEPT (SELECT * FROM destinations_rel))
- (3) INSERT INTO d_destinations_rel
(SELECT d1.From, d2.To, d1.Company
FROM d1_destinations_rel AS d1, destinations_rel AS d2
WHERE (d1.To=d2.From) AND (d1.Company=d2.Company)
UNION
SELECT d1.From, d2.To, d1.Company
FROM destinations_rel AS d1, d1_destinations_rel AS d2
WHERE (d1.To=d2.From) AND (d1.Company=d2.Company)
UNION
SELECT d1.From, d2.To, d1.Company
FROM d1_destinations_rel AS d1, d1_destinations_rel AS d2
WHERE (d1.To=d2.From) AND (d1.Company=d2.Company)
EXCEPT (SELECT * FROM d1_destinations_rel)
EXCEPT (SELECT * FROM destinations_rel)
EXCEPT (SELECT * FROM d_destinations_rel))

SQL statements (1) and (2) are executed only once, since they correspond to non recursive rules. On the contrary, the statement (3) is executed several times, until the least fixpoint is reached, i.e. `d_destinations_rel` is empty. Note that `d_destinations_rel` corresponds to one of the relations ΔR^k introduced previously, whereas `d1_destinations_rel` corresponds to ΔR^{k-1} . The

evaluation algorithm initializes `d1_destinations_rel` to `destinations_rel` at the first iteration and, then, suitably updates the tuples of `destinations_rel` and `d1_destinations_rel` from the new values derived at each iteration in `d_destinations_rel`.

3 Experiments and Benchmarks

Compared systems. The main purpose of our experiments has been to assess the efficiency of DLV^{DB} as a Deductive Database system. In order to provide a comparative and comprehensive analysis with the state-of-the-art systems in the field, we have compared DLV^{DB} performance with: (i) DB2 and SQLServer, because they are among the most efficient Database System Engines and they support the execution of SQL99 recursive statements; (ii) LDL++, because it is one of the most robust implementations of DDS; (iii) XSB, as an efficient implementation of the Top-Down evaluation strategy; (iv) Smodels, one of the most widely used Answer Set Programming systems along with DLV. In the following, we use the symbol DLV^{DB} to indicate the direct database execution, whereas we use DLV for the main memory execution of our system.

All tests have been carried out on a Pentium 4 machine with a 1.4 GHz CPU and 512 Mbytes of RAM, using Windows XP SP2 operating system for DBMSs and Linux Debian for logic-based systems.

Benchmark Problems and Data. In the past great research efforts have been carried out to define problems and data structures particularly suited for testing Deductive Database Systems. In our tests, we used the problems and the data structures introduced in [3]; these basically resort to the execution of recursive queries (Reachability and Same Generation) over graph-based data structures.

The complete list of the problem encodings exploited in our tests can be found at the address www.mat.unical.it/terracina/ismis06/encodings.pdf. It is worth pointing out that none of the considered DBMSs follows exactly the SQL99 standard for the definition of recursive queries; as a consequence, proprietary constructs have been exploited for each DBMS.

As for benchmark data structures, we considered: (i) full binary trees, (ii) acyclic graphs (a-graphs in the following), (iii) cyclic graphs (c-graphs in the following) for the Reachability problem, whereas we exploited full binary trees for Same Generation. For each data structure various instances of increasing dimensions have been constructed. Graphs have been generated using the Stanford GraphBase [9] library. The size of each data structure is measured in terms of the number of facts describing it.

For each considered problem and data set we measured the performance of the various systems in answering three kinds of queries: (i) unbound queries, (ii) queries with one parameter bound, (iii) queries with two parameters bound. We considered these three cases because DBMSs and Deductive Databases generally benefit of bounds in the queries, whereas Answer Set Programming systems are generally more effective with unbound queries; as a consequence, it is interesting to test all these systems in both their favorable and unfavorable contexts. It is

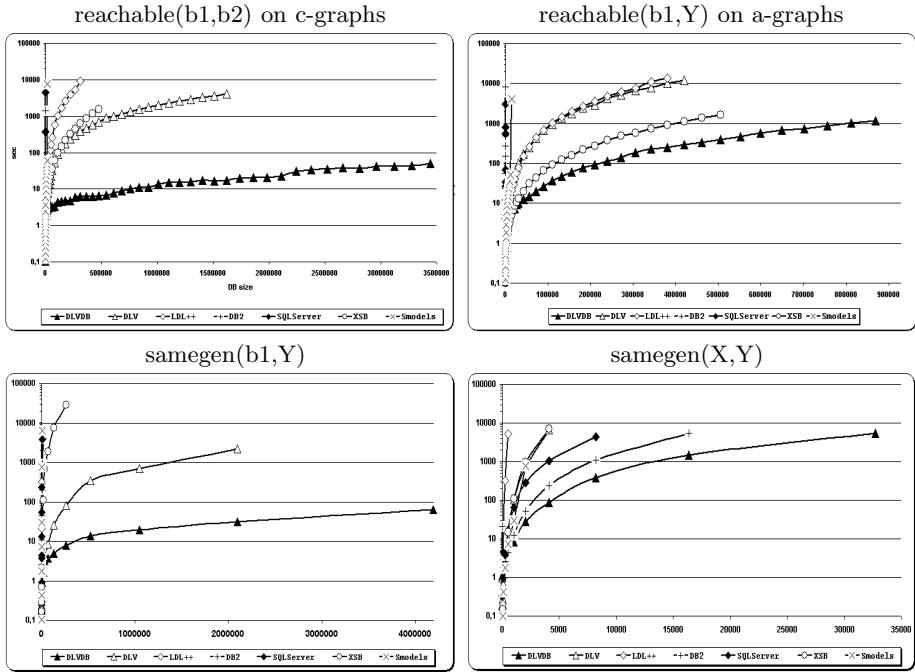


Fig. 3. Time response for some queries. Input dimension is the number of input facts

worth pointing out that some of the tested systems (e.g., DLV, DLV^{DB} and LDL++) implement optimization strategies, e.g., ‘a la magic set’ [4,11], typical of deductive databases, which automatically rewrite the input program and the query, trying to push down as much as possible constants possibly present in the query; as a consequence, the actually evaluated programs are the optimized ones automatically derived by these systems, but the cost of the rewriting is always considered in systems’ performance.

Results. Figure 3 shows obtained results for some queries; in the figure, the symbols $b1$ and $b2$ indicate constant values. Due to space constraints we cannot show here the results for all the queries; we selected the most representative ones. In these tests, we fixed a maximum running time of 12000 seconds (about 3 hours) for each test. In Figure 3 the line of a system stops whenever some query was not solved within this time limit (note that graphs have a logarithmic scale).

From the analysis of this figure we can observe that, in several cases, the difference of performance of DLV^{DB} w.r.t. almost all the other systems is in orders of magnitude; moreover, there is no system which can be considered the “competitor” of DLV^{DB} in *all* the tests. In fact, as for `reachable(b1,Y)`, the system having the best performance after DLV^{DB} is XSB, but its performance significantly decrease w.r.t. DLV^{DB} in the other problems. As for `samegen(X,Y)` we have that the only system which is competitive with DLV^{DB} is DB2, but in all the other cases its performance significantly decreases w.r.t. DLV^{DB} . For the

Table 2. Execution times of the systems capable of solving the query for the maximum considered size of the input data

Query	Data Type	Max Size (tuples)	DB2 (sec)	DLV (sec)	DLV ^{DB} (sec)	LDL++ (sec)	Smodels (sec)	SQLServer (sec)	XSB (sec)
reachable(X,Y)	tree	4194302	—	—	11161	—	—	7280	—
reachable(b1,Y)	tree	4194302	—	—	76	—	—	6438	—
reachable(b1,b2)	tree	4194302	—	—	591	—	—	12	—
reachable(X,Y)	a-graph	929945	—	—	11820	—	—	—	—
reachable(b1,Y)	a-graph	929945	—	—	1191	—	—	—	—
reachable(b1,b2)	a-graph	929945	—	—	4	—	—	—	—
reachable(X,Y)	c-graph	612150	—	—	11936	—	—	—	—
reachable(b1,Y)	c-graph	612150	—	—	11933	—	—	—	—
reachable(b1,b2)	c-graph	612150	—	981	8	—	—	—	—
samegen(X,Y)	tree	32766	—	—	5552	—	—	—	—
samegen(b1,Y)	tree	4194302	—	—	64	—	—	—	—
samegen(b1,b2)	tree	4194302	—	—	102	—	—	—	—

other queries shown in Figure 3, namely *reachable(b1,b2)* and *samegen(b1,Y)*, the difference between DLV^{DB} and all the other systems is more than one order on magnitude. Finally, DLV^{DB} is always capable of handling much larger amounts of data w.r.t. all the other systems.

We have observed these kinds of situations in almost all the tests we carried out. Interestingly enough, DBMSs often have the worst performance and they can handle very limited amounts of input data, except for some kinds of queries; as an example, SQLServer showed very good performance only for Reachability on trees (see also Table 2 introduced next). This behaviour could be justified by the presence of ad-hoc optimization mechanisms implemented in this system which, however, are not effective for other kinds of queries.

As pointed out also in [3], another important parameter to measure in this context is the system's capability of handling great amounts of input data. In order to carry out this kind of verification, we considered the time response of each system for the biggest input data set we have used in each query.

Table 2 shows the execution times measured for those systems which have been capable of solving the query within the fixed time limit of 12000 seconds. From the analysis of this table, we may observe that: (i) DLV^{DB} has been always capable of solving the query on the maximum data size; (ii) in 9 queries out of 12 DLV^{DB} (along with its main memory execution DLV) has been the only system capable of completing the computation within the time limit; (iii) only in two cases DLV^{DB} did not have the best time response, namely in *reachable(X,Y)* on trees and in *reachable(b1,b2)* on trees, where the best performing system has been SQLServer; as previously pointed out, this performance of SQLServer can be justified by ad-hoc optimization techniques implemented in it for this kind of query. In the other case, the difference between DLV^{DB} and SQLServer is in orders of magnitude.

Summarizing, the results of our experiments clearly show that DLV^{DB} outperforms the other systems in the direct database modality both for execution times and maximum sizes of handled data. In several cases, the time differences of DLV^{DB} w.r.t. the other systems for big input data are in orders of magnitude,

especially for bound queries. Finally, it presents excellent performance for all the considered queries (which are classical DDS queries) whereas other systems work well only in some specific cases.

4 Conclusions

In this paper we have presented DLV^{DB} , a new system for reasoning on massive amounts of data directly in mass-memory data structures. It presents all the features of a DDS but supports also the features of disjunctive logic programming systems. The experimental validation shows that DLV^{DB} provides both important speed up in the running time and the capability to handle larger amounts of data w.r.t. existing systems.

In the future we plan to extend the language supported by the direct database execution and to exploit the system in interesting research fields, such as the fast prototyping of integration systems.

References

1. American National Standards Institute:ANSI/ISO/IEC 9075-1999 (SQL:1999, Parts 1-5). New York, NY, 1999.
2. I. Balbin and K. Ramamohanarao. A generalization of the differential approach to recursive query evaluation. *Journal of Logic Programming*, 4,(3):259–262, 1987.
3. F. Bancilhon and R. Ramakrishnan. Performance evaluation of data intensive logic programs. In *Foundations of Deductive Databases and Logic Programming.*, pages 439–517. Morgan Kaufmann, 1988.
4. C. Beeri and R. Ramakrishnan. On the power of magic. *Journal of Logic Programming*, 10(1-4):255–259, 1991.
5. T. Dell’Armi, W. Faber, G. Ielpa, C. Koch, N. Leone, S. Perri, and G. Pfeifer. System Description: DLV. *Proc. of LPNMR’01*., LNAI, pages 409–412, Vienna, Austria. Springer Verlag, 2001.
6. T. Dell’Armi, W. Faber, G. Ielpa, N. Leone, and G. Pfeifer. Aggregate Functions in Disjunctive Logic Programming: Semantics, Complexity, and Implementation in DLV. *Proc. of IJCAI 2003*, pages 847–852, Acapulco, Mexico. 2003.
7. E. Franconi, A. Laureti Palma, N. Leone, S. Perri, and F. Scarcello. Census Data Repair: a Challenging Application of Disjunctive Logic Programming. In *In Proc. of LPAR 2001*, pages 561–578, 2001.
8. H. Garcia-Molina, J. D. Ullman, and J. Widom. *Database System Implementation*. Prentice Hall, 2000.
9. Donald E. Knuth. *The Stanford GraphBase : A Platform for Combinatorial Computing*. ACM Press, New York, 1994.
10. N. Leone, G. Greco, G. Ianni, V. Lio, G. Terracina, et al. The infomix system for advanced integration of incomplete and inconsistent data. In *Proc. of SIGMOD 2005*, pages 915–917, Baltimore, USA. ACM Press. 2005.
11. I.S. Mumick, S.J. Finkelstein, H. Pirahesh, and R. Ramakrishnan. Magic conditions. *ACM Trans. Database Systems*, 21(1):107–155, 1996.
12. J.D. Ullman. *Principles of Database and Knowledge Base Systems*. Computer Science Press, 1989.

A Logic-Based Approach to Model Supervisory Control Systems

Pierangelo Dell'Acqua^{1,2}, Anna Lombardi¹, and Luís Moniz Pereira²

¹ Department of Science and Technology - ITN
Linköping University, 601 74 Norrköping, Sweden
`{pier, annlo}@itn.liu.se`

² Centro de Inteligência Artificial - CENTRIA
Departamento de Informática, Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
`lmp@di.fct.unl.pt`

Abstract. We present an approach to model supervisory control systems based on extended behaviour networks. In particular, we employ them to formalize the control theory of the supervisor. By separating the reasoning in the supervisor and the action implementation in the controller, the overall system architecture becomes modular, and therefore easily changeable and modifiable.

1 Introduction

Hybrid control systems [8] typically arise from the interaction of discrete planning algorithms and continuous processes and they represent the basic structure of discrete event supervisory controllers for continuous systems. Hybrid control systems have been object of research for more than a decade and there are well-established results. Several approaches to hybrid control systems have been defined in the literature, see e.g. [1,6,7] where examples of the most common structures are given. In supervisory control [5], the *system* (both plant and controller) consists of a family of subsystems that can be actuated at different time instants. Controller selection is carried out by means of logic-based switching. Switching among candidate controllers is performed by a high-level decision maker called a *supervisor*. In this way one level of abstraction is introduced making explicit the reasoning performed by the supervisor. The task of the controller is to execute the action selected by the supervisor and accordingly steer the plant.

Behaviour networks were introduced by Pattie Maes [9,10] to address the problem of action selection in environments that are dynamic and too complex to be entirely predictable, and where the system has limited computational resources and time resources¹. Therefore, the action selection problem cannot be completely rational and optimal. Maes adopted the stance suggestive of building intelligent systems as a society of interacting, mindless agents, each having its

¹ See pp. 244-255 in [4] for a summary introduction.

own specific competence [2,11]. The idea is that competence modules cooperate in such a way that the society as a whole functions properly. Such an architecture is very attractive because of its distributiveness, modular structure, emergent global functionality and robustness [9]. The problem is how to determine whether a competence module should become active (i.e., selected for execution) at a certain moment. Behaviour networks addressed this problem by creating a network of competence modules and by letting them activate and inhibit each other along the links of the network. Global parameters were introduced to guide the activation/inhibition dynamics of the network. Behaviour networks combine characteristics of traditional AI and of the connectionist approach by using a connectionist computational model on a symbolic, structured representation. In a previous paper [3] we adapted the formalism of behaviour networks to make it possible to model hybrid control systems.

This paper proposes the use of extended behaviour networks to formalize the control theory of the supervisor in supervisory control systems. The reason for doing so is a consequence of the following observation. Extended behaviour networks are used in [3] to implement the controller itself. The resulting structure of the controller is rather complex as the controller has both the task of reasoning and the task of implementing the selected action. By separating the reasoning in the supervisor and the action implementation in the controller, the overall architecture becomes modular and therefore can be easily changed and/or extended. For example, if one new rule is introduced the only module that must be changed is the supervisor while the controller does not need any change.

2 Behaviour Networks

In this section we recap the approach to extended behaviour networks introduced in [3]. This approach extends the original framework of behaviour network proposed by Pattie Maes [9,10] to allow rules containing variables, internal actions, integrity constraints, and modules (sets of atoms and rules).

A behaviour network is characterized by six modules: R, P, H, C, G and E. The module R is a set of rules formalizing the behaviour of the network, P is a set containing the global parameters, H is the internal memory of the network, C its integrity constraints, G its goals/motivations, and E the module that receives the input from the environment. We call the *state* of the network the tuple $S=(R, P, H, C, G, E)$. We assume given a module Math containing the axioms of elementary mathematics.

2.1 Language

A *term* is a constant c or a variable x . Given a predicate symbol q of arity n and the terms $t_1, \dots, t_n$, then $q(t_1, \dots, t_n)$ is an *atom*. When the arity of q is 0, we write the atom as q . To express that an atom belongs to a module, we employ the notion of indexed atoms. In the following, we name the modules of the network by using small letters. Thus, we name the modules R, P, H, C, G,

E and Math as r , p , h , c , g , e and $math$. Let α be any arithmetic expression. Given a module m and an atom A , an *indexed atom* has the form $m:A$ or $m \div A$. The first states that A belongs to m , while the second states that A does not belong to m . We write sequences of indexed atoms by separating them with the symbol ', ', and we write ε to indicate the empty sequence.

Both goals and integrity constraints are sequences of indexed atoms. A goal (motivation) expresses some condition to be achieved, while an integrity constraint represents a condition that must not hold. A *rule*² is a tuple of the form:

$$\langle prec; del; add; action; \alpha \rangle$$

where *prec*, *del* and *add* are sequences (possibly empty) of indexed atoms. *prec* denotes the preconditions that have to be fulfilled before the rule can become executable. *del* and *add* represent the internal effect of the rule in terms of a delete and add sequence of indexed atoms. When both *del* and *add* are empty (i.e., ε), then the rule has no internal effects. We assume that *del* and *add* contain only indexed atoms of the form $m:A$ with $m \neq e$ and $m \neq math$. The reason for this restriction is that the modules E and Math cannot be updated. In contrast, *prec* can contain any indexed atom.

The atom *action* represents the external effect of the rule: an action that must be executed. We employ 'noaction' to indicate that the rule does not have any external effect. Finally, each rule has a level α of strength³. Variables in a rule are universally quantified over the entire rule.

At every state S , a rule in R must be selected for execution. To do so, one needs to find all the rules that are executable and select one. A framework of rule selection for behaviour network is discussed in [3].

3 Case Study: Modelling an Artificial Animat

In this section we illustrate how to model a virtual animat by means of a supervisory control system. The supervisor of the system, on the basis of the input and output signals of the process, chooses which action to execute. Then the system activates the controller implementing the action selected by the supervisor. Thus, selecting an action to be executed corresponds in supervisory control system terms to select and activate a controller. We employ behaviour networks as a formalism (for the supervisor) to describe controller selection.

Scenario

Consider a virtual marine world inhabited by a variety of fish [12]. They autonomously explore their dynamic world in search for food. Fishes are situated within the environment, and sense and act over it. For simplicity, the behaviour of a fish is reduced to eating food, flocking and escaping from predators and is determined by the motivation of it being safe and satiated.

² In [10] rules are called competence modules.

³ This value is used to calculate the activation level of the rules.

Supervisor

At every time instant t_k the supervisor receives information on the state of the fish, for example $hungry(t_k)$, $fear(t_k)$ and $distance\_obstacle(t_k)$. Some of the actions that the supervisor can select are: *searchFor(food)*: to search for food when it is hungry; *flee*: to run away from danger; and *flock*: to move in a flock. Assume that there exists a constraint $C=\{e:fear(x), math:x>5, h:searching(food)\}$ needed to prevent the fish to start searching for food in dangerous situations. The module G is $\{h:safe, h:satiated\}$, and R contains (among others) the following rules.

```

(e:fear(x),m:x<0.2,h÷safe; ε; h:safe; noaction; 0.4)

(e:fear(x),m:x>0.5,h:flocking; h:safe,h:flocking; fleeing; flee(x*2);
x*2)

(e:fear(x),m:x>0.5,h:alone; h:safe,h:searching(food); fleeing; flee(x*5);
x*5)

```

The first rule represents an internal action. It states that if the fish does not have fear, then it will add to its mental state that it is in a safe condition. Instead, the next two rules state that if the fish has fear, then it will flee with different values depending on whether it is flocking or alone. The level of strength of the last two rules is directly proportional to the amount of fear. The next three rules characterize the behaviour of the fish in case it gets hungry, and when it decides to flock.

```

(e:hungry(x),m:x>0.5; h:satiated; h:searching(food); search(food); 0.5)

(e:hungry(x),m:x<0.2; h:searching(food); h:satiated; noaction; 0.5)

(h÷searching(food); h:alone; h:flocking; flock; 0.4)

```

The last rule above has a non-monotonic flavour (via $\div$). It says that if the fish is not searching for food, then it will start flocking. This is needed to give continuity to the fish behaviour.

Controller

The controller module contains all the controllers. Each controller implements an action selected by the supervisor. In the fish scenario, depending on the chosen action, the controller calculates the direction and velocity of the fish. For example, if the supervisor selects a rule whose action is *flee*(0.6), then the corresponding controller can be described as:

```

flee(fleeFactor) {
   $\vec{v} = \text{fleeFactor} * \sum_{n=1}^N \frac{\vec{x}_{fish} - \vec{x}_n}{\|\vec{x}_{fish} - \vec{x}_n\|};$ 
  return  $\vec{v}$ ;
}

```

where N is the total number of visible predators, and $\vec{x}_n$ is the vector of the position of the n -th predator. We write $\|\vec{x}\|$ to indicate the norm of a vector

$\vec{x}$. When a predator comes too close to the fish, the controller `flee` makes the fish flee in the opposite direction. The new direction is calculated wrt. the visible predators. `fleeFactor` determines the strength of the willing of the fish to escape. The velocity $\vec{v}$ is then passed to the plant where the new position will be calculated.

Plant

The artificial fish is modelled by several state variables representing the stimuli of the fish: e.g., the stimuli of hunger and fear. They are represented as variables with values in the range $[0\ 1]$ with higher values indicating a stronger desire to eat or to avoid predators [12]:

- *hunger*: it expresses how hungry the fish is and it is approximated by:

$$H(t) = \min \{ \Delta T \cdot \alpha, 1 \} \quad (1)$$

where ΔT denotes the time since the last meal, and α indicates the appetite of the fish.

- *fear*: it quantifies the fear of the fish by taking into account the distance $d(t)$ of the fish to a predator:

$$D(t) = \min \{ D_0/d(t), 1 \} \quad (2)$$

where D_0 indicates how brave the fish is: if the fish is brave, it can take some risk by coming close to the danger, for example to reach the food.

Any time the plant receives the velocity $\vec{v}$ from a controller, the new position of the fish is calculated:

$$\vec{x}_{fish}(t+h) = \vec{x}_{fish}(t) + \vec{v} \cdot h$$

where h is the time interval between two computations. Finally, the new values for the fish stimuli are updated wrt. the new position $\vec{x}_{fish}(t+h)$.

4 Future Work

An interesting extension to the language L of the behaviour network would be to allow variables to occur in the strength levels of rules. This will allow defining the strength of a rule as a function of state.

One may also consider the possibility of including preference rules into a behaviour network with the aim of contributing to the action selection process. The idea being to compute all the executable rules whose activation level is above a certain threshold, and then to use preference reasoning to select the one that becomes active (and not just the one with greatest activation). This will allow for additional flexibility, via a new threshold parameter, and an extra level of control by means of context sensitive preferences.

Our method affords a clear declarative, modular, and updatable rule based enactment of control, both at definitional and implementation levels, by separating rules for controller choice from the controllers themselves and the setting of their parameters. Flexibility and strategic leverage is gained in so doing, paving the way for controller monitoring, rule and parameter self-updating, and multi-level control. Furthermore, behaviour networks permit a more reactive or more deliberative form of controller choice, depending on the combination of the relative strengths of top-down and bottom-up "energy" propagations. In the future, meta-rules for updating the controller choice rules will be possible, either explicitly given or formed on the basis of learning or of pro-activeness through simulation.

References

1. Panos J. Antsaklis and Anil Nerode. Hybrid control systems: An introductory discussion to the special issue. *IEEE Transactions on Automatic Control*, 43(4):457–460, April 1998. Guest Editorial.
2. R. A. Brooks. A robust layered control system for a mobile robot. *IEEE J. of Robotics and Automation*, 2(1):14–23, 1986.
3. P. Dell’Acqua, A. Lombardi, and L. M. Pereira. Modelling Hybrid Control Systems with Behaviour Networks. In J. Filipe, J.-L. Ferrier, and J. A. Cetto, editors, *2nd Int. Conf. on Informatics in Control, Automation and Robotics (Icinco05). Procs. Intelligent Control Systems and Optimization Vol.1*, pages 98–105. INSTICC Press. ISBN:972-8865-29-5, 2005.
4. Stan Franklin. *Artificial Minds*. MIT Press, 1995.
5. Joao P. Hespanha, Daniel Liberzon, and A. Stephen Morse. Overcoming the limitations of adaptive control by means of logic-based switching. *Systems and Control Letters*. To appear.
6. W. Kohn and A. Nerode. Models for hybrid systems: automata, topologies, controllability and observability. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, LNCS 736, pages 317–356, Berlin, 1993. Springer-Verlag.
7. W. Kohn and A. Nerode. Multiple agent hybrid control architecture. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, LNCS 736, pages 297–316, Berlin, 1993. Springer-Verlag.
8. Feng Lin. Robust and adaptive supervisory control of discrete event systems. *IEEE Transactions on Automatic Control*, 38(12):1848–1852, December 1993.
9. Pattie Maes. How to do the right thing. *Connection Science Journal, Special Issue on Hybrid Systems*, 1(3):291–323, 1989.
10. Pattie Maes. A bottom-up mechanism for behavior selection in an artificial creature. In J. A. Meyer and S. Wilson, editors, *Proceedings of the first International Conference on Simulation of Adaptive Behavior*. MIT Press, 1991.
11. M. Minsky. *The Society of Mind*. Simon and Schuster, New York, 1986.
12. Xiaoyuan Tu. *Artificial Animals for Computer Animation: Biomechanics, Locomotion, Perception, and Behavior*. PhD thesis, ACM Distinguished Ph.D Dissertation Series, LNCS 1635, 1999.

SAT as an Effective Solving Technology for Constraint Problems

Marco Cadoli, Toni Mancini, and Fabio Patrizi

Dipartimento di Informatica e Sistemistica
Università di Roma “La Sapienza”

Via Salaria 113, I-00198 Roma, Italy

{cadoli, tmancini, patrizi}@dis.uniroma1.it

Abstract. In this paper we investigate the use of SAT technology for solving constraint problems. In particular, we solve many instances of several common benchmark problems for CP with different SAT solvers, by exploiting the declarative modelling language NPSPEC, and SPEC2SAT, an application that allows us to compile NPSPEC specifications into SAT instances. Furthermore, we start investigating whether some reformulation techniques already used in CP are effective when using SAT as solving engine. We present encouraging experimental results in this direction, showing that this approach can be appealing.

1 Introduction

Several benchmark problems have been proposed in the literature for testing the performance of Constraint Programming tools e.g., OR-Library (www.ms.ic.ac.uk/info.html) or CSPLib (www.csplib.org), spanning several areas, from combinatorics, to planning, scheduling, or problems on graphs. There is also a great variety in the kind of solvers that are used for Constraint Programming: those based on backtracking (cf., e.g., [15]), mathematical programming (cf., e.g., [5]), answer sets and stable model semantics (cf., e.g., [8]), which are typically complete, or those that rely on local-search techniques (cf., e.g., [10]) which are intrinsically incomplete.

In this paper we focus on a different kind of tools, i.e., solvers for Propositional Satisfiability (SAT), showing how they can be transparently and effectively used for solving constraint problems. The intuition behind the usage of a SAT solver, is that every CSP can be reduced in polynomial time to an instance of SAT, since the complexity of solving a CSP is in NP, and SAT is one of the prototypical NP-complete problems. Actually, the latter aspect led to a great interest and to a huge amount of research in the field of SAT solving (cf., e.g., the proceedings of the last SAT conferences), leading to the current availability of very efficient solvers that can deal with very large formulae. State-of-the-art SAT solvers include complete ones, such as zCHAFF [11], and incomplete ones, such as WALKSAT [14], and BG-WALKSAT [16]. For an up-to-date list, we refer the reader to the URLs www.satlib.org and www.satlive.org.

The availability of fast solvers for SAT drew a great interest in the CP research community, and many papers show how to translate (compile) into SAT instances of various problems (cf., e.g., [7,6]). However, the complexity of the translation task is a major obstacle, since the compilation strongly depends on the constraints of the problem to be solved. Nowadays, this task is typically made by problem-dependent programs hence, in practice, preventing SAT to be one of the actual solving technologies for Constraint Programming. The availability of specification languages and systems that compile problem instances into SAT formulae, e.g., the language NPSPEC and the SPEC2SAT system [3], is an important step ahead, providing the user with the possibility of easily building specifications for new constraint problems in a purely declarative way, maintaining a strong independence on the instances.

In this paper, we present some experiments showing that SAT technology can be *effectively* employed for solving CSPs, by using NPSPEC on several common benchmark problems for CP, experiencing different SAT solvers. The problems we focus on are a significant subset of those present in the benchmark repository CSPLib, very well-known in the CP research community. Problems in CSPLib are usually described only in natural language, and no formal specification is given for most of them. Hence, as a side-effect, our work also proposes declarative specifications (in the language NPSPEC) for such problems.

In general, given a specification of a problem, several techniques have been proposed to reformulate it, in order to improve the solver efficiency, while maintaining equivalence (or at least, the possibility to efficiently reconstruct valid solutions to the original problem from solutions to the reformulated one). We experienced the application, at the symbolic level of the specification, of two of them, and present some experimental results.

The paper is organized as follows: in Section 2 we briefly illustrate the language NPSPEC and the SPEC2SAT program that, given a NPSPEC specification and an instance, compiles it into a SAT instance. In Section 3 we present the chosen benchmark problems, while in Section 4 we present and comment our experimental results. Finally, Section 5 draws further discussions and concludes the paper.

2 The NPSpec Language and Spec2Sat

NPSPEC and SPEC2SAT have been extensively described in [3]. Hence, in what follows, we just recall the syntax and the informal semantics of the modelling language, and the general architecture of the compiler. The Home Page of the NPSPEC project (www.dis.uniroma1.it/~cadoli/research/projects/NP-SPEC/) contains all the specifications proposed in this paper, as well as the program itself.

The NPSPEC language. An NPSPEC program consists of a **DATABASE** section and a **SPECIFICATION** section. The former includes the definition of the problem instance, in terms of extensional relations, and integer intervals and constants. The latter section instead, consists of the problem specification, that is divided

into two parts: the declaration of a *search space*, and the definition of constraints that a point in the search space has to satisfy in order to be a solution to the problem instance. The declaration of the constraints is given by a stratified DATALOG program, which can include the six predefined relational operators and negative literals.

The full syntax of NPSPEC is given in [3], hence here we just recall it with an example. In particular, we show an NPSPEC program for the *Social golfer* NP-complete problem (problem nr. 10 in CSPLib), that, given a set of `N_PLAYERS` players that want to play golf once a week, amounts to find an arrangement for all of them into a number `N_GROUPS` of groups of size `GROUP_SIZE` for `N_WEEKS` weeks, in such a way that no two players play more than once in the same group. The following relationship must hold among three of the afore-mentioned quantities: $N\_PLAYERS = N\_GROUPS * GROUP\_SIZE$.

DATABASE

```
N_WEEKS = 3;   N_GROUPS = 3;   GROUP_SIZE = 3;
N_PLAYERS = N_GROUPS * GROUP_SIZE;
```

SPECIFICATION

```
IntFunc({1..N_PLAYERS})<<{1..N_WEEKS}, play, 1..N_GROUPS). // S1
fail <-- play(P1,W1,G1), play(P2,W1,G1), P1 != P2,           // S2
         play(P1,W2,G2), play(P2,W2,G2), W1 != W2.
fail <-- COUNT(play(*,W,G),X), X != GROUP_SIZE.              // S3
```

The following comments are in order:

- The input instance is defined in the `DATABASE` section, which is generally provided in a separate file (this part may also define finite relations).
- In the search space declaration (metarule `S1`) the user declares the predicate symbol `play` to be a “guessed” one, implicitly of arity 3. All other predicate symbols are, by default, not guessed. Being guessed means that we admit all extensions for the predicate, subject to the other constraints.
- `play` is declared to be an integer function from the finite domain $\{1..N\_PLAYERS\} \times \{1..N\_WEEKS\}$ to $\{1..N\_GROUPS\}$. As an example, $\{(1, 1, 1), (2, 1, 1), (3, 1, 1), (4, 1, 2), (5, 1, 2), (6, 1, 2), (7, 1, 3), (8, 1, 3), (9, 1, 3), \dots\}$ is a valid extension as long as it defines exactly one group for any player in any week.
- Comments can be inserted using the symbol “//”.
- Rules `S2` and `S3` are the constraints that schedulings must obey in order to be valid solutions: a scheduling *fails*, i.e., it is not valid, if there exist two players `P1` and `P2` that play twice in the same group (rule `S2`), or if there exist a group `G` which is not of size `GROUP_SIZE` in a week `W` (rule `S3`).

Solving this program with NPSPEC produces an extension for the guessed predicates that satisfies all the constraints, if it exists.

The search space declaration, which corresponds to the definition of the domain of the guessed predicates, is, in general, a sequence of declarations of the form: (i) `Subset(<domain>, <pred_id>);` (ii) `Permutation(<domain>, <pred_id>);` (iii) `Partition(<domain>, <pred_id>, n);`

(iv) `IntFunc(<domain>, <pred_id>, min..max)`. We do not formally give further details of the NPSPEC syntax, but, in the following sections, we present and comment several other examples. We just remark that the declarative style of programming in NPSPEC is very similar to that of DATALOG, and it is therefore easy to extend programs for incorporating further constraints. Concerning syntax, we remark that NPSPEC offers also useful SQL-style aggregates, such as SUM, COUNT, MIN, and MAX. Several examples of Section 3 use such operators.

The NPSPEC to SAT compiler. SPEC2SAT is an application that allows the compilation of a NPSPEC specification (when given together with input data) into a SAT instance. We do not give here the technical details of the compilation task, that can be found in [3], but just briefly describe the general architecture of the application. Given two text files, containing the specification S in NPSPEC and the instance data I , SPEC2SAT compiles $S \cup I$ into a CNF formula T in DIMACS format, and builds an object representing a dictionary which makes a 1-1 correspondence between ground atoms of the Herbrand base of $S \cup I$ and propositional variables of the vocabulary. The file in DIMACS format is given as an input to a SAT solver (the choice of the SAT solver is completely independent of the application, as long as it accepts the standard DIMACS format as input, and can be chosen by the user), which delivers either a model of T , if satisfiable, or the indication that it is unsatisfiable. At this point, the MODEL2SPEC module performs, using the dictionary, a backward translation of the model (if found) into the original language of the specification.

3 Benchmark Problems

As mentioned in Section 1, in this paper we show results in solving typical Constraint Programming benchmark problems using SAT technology, by taking advantage of SPEC2SAT. In this section, we list the problems used for the experiments, that are a significant subset of those present in the well-known library CSPLib (www.csplib.org). We chose eight problems which cover five out of the seven classes of problems offered by the CSPLib. As a side-effect, we obtain declarative specifications (in the language NPSPEC) for such problems. Due to lack of space, we describe (apart for Social golfer, presented in Section 2) the NPSPEC specifications of only few of them. The others (and several more) can be found on-line.

Golomb rulers (problem nr. 006). A Golomb ruler of length L with m marks is defined as a set of m integers $0 = a_1 < a_2 < \dots < a_m = L$ such that the $m(m-1)/2$ differences $a_j - a_i$, $1 \leq i < j \leq m$ are all distinct. In our formulation, given L and m , we are interested in finding a Golomb ruler of any length *not greater than* L . An NPSPEC specification for this problem is as follows:

```
IntFunc({1..N_MARKS}, ruler, 0..LENGTH).           // G1
fail <-- NOT ruler(1,0).                             // G2
fail <-- ruler(I,V_I), ruler(J,V_J), J > I, V_I >= V_J. // G3
fail <-- ruler(I,V_I), ruler(J,V_J), ruler(K,V_K), ruler(L,V_L), // G4
```

```

      I < J, K < L, I != K, V_J - V_I == V_L - V_K.
fail <-- ruler(I,V_I), ruler(J,V_J), ruler(K,V_K), ruler(L,V_L), // G5
      I < J, K < L, J != L, V_J - V_I == V_L - V_K.

```

The search space is defined (metarule **G1**) as the set of all total integer functions from $\{1..N\_MARKS\}$ to the integer range $\{0..LENGTH\}$, hence assigning a position on the ruler to each mark. Rule **G2** forces the first mark to be at the beginning of the ruler (position 0). Rule **G3** forces marks to be in ascending order, i.e., the second mark is on the right of the first one, the third on the right of the second, and so on. Rules **G4** and **G5** force distances between two marks to be all different.

All-interval series (problem nr. 007). Given the twelve standard pitch-classes (c, c#, d, ...), represented by numbers 0,1,...,11, find a series in which each pitch-class occurs exactly once and in which the musical intervals between neighboring notes cover the full set of intervals from the minor second (1 semitone) to the major seventh (11 semitones). Here, we generalize the problem by replacing the twelve standard pitch-classes with an arbitrary set `pitch` of N pitch-classes (all-interval series problem of size N). An ad-hoc encoding of this problem into SAT has been made in [6]. In NPSPEC, the All-interval series problem can be specified as follows:

```

Permutation(pitch, series). // A1
Permutation(interval, neighbor). // A2
fail <-- series(P,X), series(Q, X+1), NOT neighbor(abs(P-Q), X). // A3

```

By metarule **A1**, guessed predicate `series` assigns a different order number to every pitch in the series. Metarule **A2** guesses a second guessed predicate, `neighbor`, to be an ordering of all possible intervals (`interval` is defined in the instance file to be the integer range $[1, N - 1]$): a tuple $\langle intv, idx \rangle$ in `neighbor` means that pitches at positions idx and $idx + 1$ in the series are divided by interval $intv$. The final rule **A3** actually forces `series` to respect this constraint: for each pair of adjacent pitches P and Q (at positions X and $X+1$), the interval dividing them must have order X in the permutation `neighbor`.

Schur's lemma (problem nr. 015). The problem is to put N_BALLS balls labelled $\{1, ..., N\_BALLS\}$ into N_BOXES boxes so that for any triple of balls (x, y, z) with $x + y = z$, not all are in the same box. An NPSPEC specification for this problem is as follows:

```

Partition({1..N_BALLS}, putIn, N_BOXES). // S1
fail <-- putIn(X,Box), putIn(Y,Box), putIn(Z,Box), X + Y == Z. // S2

```

Metarule **S1** declares the search space to be the set of all partitions of the set of N_BALLS balls into N_BOXES boxes, while rule **S2** expresses the constraint.

The other problems we considered are *Ramsey* (nr. 017) *Magic square* (nr. 019), *Langford's number* (nr. 024), and *Balanced academic curriculum* (BACP, nr. 030).

4 Experiments

For all problems we chose a non-trivial set of instances – by using, when possible, publicly available benchmarks (e.g. for BACP) – and compiled them into SAT. Then, we ran different SAT solvers on those instances, and measured their solving times. We used two recent SAT solvers, very different in nature: (i) zCHAFF, one of the fastest, complete solvers today available and (ii) BG-WALKSAT, a sound but incomplete one, based on local search. BG-WALKSAT is a recent extension of the well-known WALKSAT, where the search is guided by *backbones* of the formula. Furthermore, we investigated whether the application of different reformulation techniques was suitable for improving solvers' performances. In particular, we applied two, in some sense, complementary techniques: adding symmetry-breaking constraints and neglecting *safe-delay* constraints.

Symmetry-breaking. The presence of symmetries in CSPs has been widely recognized to be one of the major obstacles for their efficient resolution. To this end, different approaches have been followed in the literature in order to deal with them, the best known being that of adding proper –i.e., symmetry-breaking– constraints to the CSP model (cf., e.g., [13,4]). Along the lines of [9], we added symmetry-breaking constraints at the specification level, hence breaking “structural” symmetries (i.e. those that depend on the problem structure, and not on the particular instance considered). Such approach has been proved to be effective for different classes of solvers, on different problems, and comes natural when using a purely declarative modelling language.

Safe-delay constraints. Given a specification, a safe-delay constraint is a constraint whose evaluation can be safely ignored in a first step, hence simplifying the problem, and efficiently reinforced in a second step, when a solution to the relaxed problem has been found [2]. The importance of safe-delay constraints is that their reinforcement can always be done in polynomial time, without further search. Highlighting and delaying safe-delay constraints can be very effective because: (i) The set of solutions is enlarged, and this can be beneficial for some solvers; (ii) The instantiation can be more efficient, since fewer constraints have to be grounded: this of course applies also to the SAT case, where delaying constraints reduces the number of generated clauses; (iii) The reinforcement of delayed constraints is often very efficient, e.g., linear or logarithmic time in the size of the input (cf. examples below). It is worth noting that also the deletion of safe-delay constraints is done by reformulating the declarative specification of the problem, hence independently of the instance.

Right now, all the reformulations have been performed manually. However, in previous work [2,9,1], we showed how the required forms of reasoning can be in principle autonomously made by system, since they reduce to check properties of first-order formulae.

For the problems presented in Section 3, we considered the following instances:

- Golomb rulers: lengths up to 15, and numbers of marks up to 9;
- All-interval: pitch classes up to 18 (zCHAFF) and to 40 (BG-WALKSAT);

Table 1. Results of the experiments using zCHAFF (a) and BG-WALKSAT (b)

Problem name	zCHAFF					
	Instances			SAT compil time (sec)	SAT solving time (sec)	Total time (sec)
	nr.	solved	unsolved			
Golomb Ruler	34	34	0	39412.96	2.46	39415.42
	34	34	0	40366.67	2.82	40369.49
	34	34	0	26654.29	27.66	26681.95
All-Interval Series	14	13	1	6.29	6600.70	6606.99
	14	10	4	6.55	17265.64	17272.19
Social Golfer	168	110	58	66171.70	212527.78	278699.48
	168	162	6	62774.72	25185.60	87960.32
Schur's Lemma	164	164	0	2412.57	0.08	2412.65
	164	164	0	2510.13	0.12	2510.12
	164	164	0	2537.14	0.08	2537.22
Ramsey problem	85	82	3	155.24	10803.04	10958.28
	85	82	3	153.95	10802.61	10956.56
	85	82	3	154.64	10802.49	10957.13
Magic Square	3	3	0	281.16	128.59	409.75
	3	3	0	282.03	38.25	320.28
Langford's number	43	39	4	1982.14	18109.22	20091.36
	43	39	4	1983.91	17422.39	19406.3
BACP	2	2	0	2041.13	1.11	2042.24

(a)

Problem name	BG-WALKSAT				
	Instances		SAT compil time (sec)	SAT solving time (sec)	Total time (sec)
	nr.	success ratio			
Golomb Ruler	20	100%	15274.17	3528.55	18802.72
	20	100%	16285.75	4632.28	20918.03
	20	60%	7617.11	6315.08	13932.19
All-Interval Series	36	17%	171.21	702.98	874.19
	36	14%	172.51	689.2	861.68
Social Golfer	137	43%	16453.92	3633.48	20087.40
	137	46%	17132.50	3792.73	20925.23
Schur's Lemma	164	100%	2412.57	4.32	2416.89
	164	99%	2510.00	4.18	2514.18
	164	100%	2537.14	7.03	2544.17
Ramsey problem	85	94%	155.24	8.47	163.71
	85	100%	153.95	7.49	161.44
	85	94%	154.64	8.05	162.69
Magic Square	3	33%	281.16	31.97	313.13
	3	33%	282.03	32.07	314.10
Langford's number	33	76%	549.76	355.53	905.29
	33	67%	549.74	357.63	907.37

(b)

- Social golfer: up to 36/3/6, or up to 25/6/5 (players/weeks/groups);
- Schur's lemma: up to 50 balls and 10 boxes;
- Ramsey problem: up to 19 nodes and 7 colors;
- Magic squares: sizes up to 4;
- Langford's number: up to 4 sets and 14 numbers;
- BACP: 2 benchmark instances, taken from CSPLib, solved with zCHAFF.

Results of our experiments are shown in Table 1, where (a) and (b) refer to zCHAFF and BG-WALKSAT, respectively, and list the overall times for compiling into SAT and solving the whole set of instances for each problem. Column “Total time” gives the gross time needed for processing the whole set of instances, summing up times for compilation and solving. Both stages had a timeout of

1 hour per instance. Finally, column “nr.” contains the number of instances run (for some problems, we were unable to solve the largest instances with BG-WALKSAT, obtaining a *too-many-clauses* error: for this reason, in Table 1(b) the number of instances is sometimes smaller), column “solved” (“unsolved”) shows how many instances have been successfully (unsuccessfully) processed within the timeout, either if sat or unsat (unsolved instances contribute with 3600 secs to the total time); since BG-WALKSAT is an incomplete solver, column “success ratio” (in (b)) reports the percentage of instances for which the correct answer is given. In addition, rows “with safe delay” and “with symmetry breaking” show the behavior of the two solvers when performing reformulation. In particular, we reformulated the following problems by safe-delay:

- Golomb rulers: rule G3 can be ignored, enlarging the set of solutions by all their permutations. However, in this case a simple modification of the other constraints is required [2]: in particular, the *absolute values* of distances between marks have to be different.
- Schur’s lemma: we let balls to be put in more than one box at the same time. If such a solution exists, a valid solution of the original problem can be derived by arbitrarily choosing a single box for each ball.
- Ramsey problem: we let multiple colors to be assigned to the same node. If such a coloring exists, then it suffices to arbitrarily choose an arbitrary color for each node having multiple ones.

In the last two cases, ignoring safe-delay constraints yields to guess multi-valued functions for the guessed predicates. This task can be accomplished by the current implementation of SPEC2SAT by defining a guessed predicate as a **multivalued** partition or integer function. We also observe that for all three problems, the second stage, i.e., recovering a solution to the original problem from a solution to the simplified one, can be performed very efficiently: in $m \log m$ for Golomb rulers (by ordering marks), and in linear time for both Ramsey and Schur’s lemma.

Concerning symmetry-breaking, we broke some of the symmetries in all but one problems. We proceeded as follows: in Golomb rulers we forced the distance between the first two marks to be less than that between the last two; in All-interval and Langford problems we forced, respectively, the first pitch-class and the first number to be less than the last one. As for Social golfer, we fixed the scheduling for the first and partially the second week, and assigned the first player always to the first group. Finally, as for Ramsey, Schur’s lemma and Magic square, we fixed the choice for the first ball/edge/square, respectively.

Some comments on the results in Table 1 are in order. First of all, both SAT solvers behave well in many cases, being able to solve instances of reasonable size. However, it is not the case that one solver is always better than the other: ZCHAFF seems much faster than BG-WALKSAT in solving Golomb rulers or BACP instances (BG-WALKSAT was not able to run on instances of the latter problem, due to their large size), while the latter performs better on All-intervals series, Ramsey, and Social golfer, even if its success ratio is not always very high. Interestingly, applying the two reformulation techniques sometimes greatly helps

ZCHAFF. As an example, by ignoring safe-delay constraints on Golomb Rulers, the overall compilation time falls down of about 13000 seconds, while the solving time increases only of about 25 seconds. A similar behavior happens also when solving this problem with BG-WALKSAT.

ZCHAFF is also positively affected by symmetry-breaking. As for Magic square and Social golfer, the speed-up is impressive. It is interesting to observe that the compilation times do not grow significantly, since the symmetry-breaking constraints we chose are quite simple. On the other hand, Schur's Lemma and Ramsey do not show any improvement. Intuitively, this is due to the lack of unsatisfiable instances in the considered set. In fact, it is well known that symmetry-breaking produces its best effects on unsatisfiable instances, avoiding the search engine to explore the whole search space before terminating with a negative answer. Unfortunately, for Schur's lemma and Ramsey, no negative instances have been found that could be handled within the timeout. Another interesting aspect is that the local search solver BG-WALKSAT does not take benefit from symmetry-breaking, hence confirming the intuition that a reduction of the solution density is an obstacle for local search [12,9].

5 Discussion and Conclusions

In this paper, we discussed how the availability of specification languages for constraint problems that automatically compile instances into SAT, can make SAT solving technology an effective tool for Constraint Programming. We experienced NPSPEC and SPEC2SAT on several well-known benchmark problems for CP, and demonstrated how they can be easily formulated in NPSPEC, and solved by exploiting state-of-the-art SAT solvers. Additionally, we showed how applying reformulation techniques such as ignoring safe-delay or adding symmetry-breaking constraints to the specifications can be very effective in improving solvers' performances. Experiments also show that the most critical point when using SAT as an engine for solving constraint problems is the compilation task, which can be, in some cases, very expensive (cf. Table 1). However, often this is compensated by the much higher (and continuously improving) efficiency of SAT solvers wrt, e.g., CP ones. In particular, experiments that have been carried out involving the state-of-the-art CP solver OPL [15] (a detailed discussion will appear in the full paper) show that:

- For some problems, e.g., Golomb rulers, Schur's lemma, Magic square, CP is much faster (e.g., as for Schur's lemma, the overall time needed by OPL is 52 seconds, against 2412 seconds needed by SPEC2SAT+ZCHAFF). However, in these cases, the real bottleneck is actually the compilation step (e.g., the actual time needed by ZCHAFF to solve all instances of such problem is only 0.08 seconds).
- For others, e.g., All-Interval, Langford's number, and Ramsey, the time needed by OPL is higher than that needed by SPEC2SAT+ZCHAFF (8328 vs. 6606, 22100 vs. 20091, 18030 vs. 10958 secs, respectively). Interestingly, the OPL models for the first two problems exhibit global constraints (*all-different*

and *distribute*, respectively), that are one of the most important features of CP solvers wrt SAT, and that may greatly improve their performance.

In our opinion, such behavior suggests that SAT can undoubtedly be considered a respectable candidate for competing with CP on combinatorial problems, and that research on more efficient techniques for compiling into SAT is a real challenge. The existence of reformulation techniques, like safe-delay, that can have a positive impact on compilation times, and recent results in related fields like Answer Set Programming suggest promising lines of research.

References

1. M. Cadoli and T. Mancini. Using a theorem prover for reasoning on constraint problems. In *Proc. of AI*IA 2005*, v. 3673 of *LNAI*, pp. 38–49, 2005.
2. M. Cadoli and T. Mancini. Automated reformulation of specifications by safe delay of constraints. *Artif. Intell.*, 2006. To appear.
3. M. Cadoli and A. Schaerf. Compiling problem specifications into SAT. *Artif. Intell.*, 162:89–120, 2005.
4. J. M. Crawford, M. L. Ginsberg, E. M. Luks, and A. Roy. Symmetry-breaking predicates for search problems. In *Proc. of KR'96*, pp. 148–159, 1996. Morgan Kaufmann.
5. R. Fourer, D. M. Gay, and B. W. Kernigham. *AMPL: A Modeling Language for Mathematical Programming*. Intl. Thomson Publ., 1993.
6. H. H. Hoos. *Stochastic Local Search - Methods, Models, Applications*. PhD thesis, TU Darmstadt, 1998.
7. H. Kautz and B. Selman. Pushing the envelope: Planning, propositional logic, and stochastic search. In *Proc. of AAAI'96*, pp. 1194–1201, 1996. AAAI Press/The MIT Press.
8. N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. The DLV System for Knowledge Representation and Reasoning. *ACM Trans. on Comp. Logic*. To appear.
9. T. Mancini and M. Cadoli. Detecting and breaking symmetries by reasoning on problem specifications. In *Proc. of SARA 2005*, v. 3607 of *LNAI*, pp. 165–181, 2005. Springer.
10. L. Michel and P. Van Hentenryck. Localizer. *Constraints*, 5(1):43–84, 2000.
11. M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an Efficient SAT Solver. In *Proc. of DAC 2001*, pp. 530–535, 2001. ACM Press.
12. S. D. Prestwich and A. Roli. Symmetry breaking and local search spaces. In *Proc. of CPAIOR 2005*, v. 3524 of *LNCIS*, pp. 273–287, 2005. Springer.
13. J.-F. Puget. On the satisfiability of symmetrical constrained satisfaction problems. In *Proc. of ISMIS'93*, v. 689 of *LNCIS*, pp. 350–361, 1993. Springer.
14. B. Selman, H. A. Kautz, and B. Cohen. Local search strategies for satisfiability testing. In *Proc. of DIMACS Challenge on Cliques, Coloring, and Sat.*, 1993.
15. P. Van Hentenryck. *The OPL Optimization Programming Language*. The MIT Press, 1999.
16. W. Zhang, A. Rangan, and M. Looks. Backbone guided local search for maximum satisfiability. In *Proc. of IJCAI 2003*, pp. 1179–1186, 2003. Morgan Kaufmann.

Dependency Tree Semantics

Leonardo Lesmo and Livio Robaldo

Dipartimento di Informatica - Università di Torino, Italy
{lesmo, robaldo}@di.unito.it

Abstract. This paper presents Dependency Tree Semantics (DTS), an underspecified logic for representing quantifier scope ambiguities. DTS features a direct interface with a Dependency grammar, an easy management of partial disambiguations and the ability to represent branching quantifier readings. This paper focuses on the syntax of DTS, while does not take into account the model-theoretic interpretation of its well-formed structures¹.

1 Introduction

The key idea of underspecification (see [3] for an overview) is to devise a formalism which allows to represent all logical readings of a sentence in a single compact structure. Such a formalism allows one to preserve compositionality without artfully casting pure semantic ambiguities into syntactical ones, as in Montagovian standard approach, but, contrary to standard logics, it yields ambiguous formulae which have to be fully disambiguated afterwards.

However, this turns into an advantage in most cases; in fact, in real contexts, not all knowledge needed for disambiguating is available during the processing of a sentence. In such cases, by using a standard logic, we should store all possible fully-specified interpretations and, when more knowledge is available, check their respective consistency in parallel. On the contrary, the use of an underspecified formalism enables a better management of incomplete information provided that such a formalisms allows to reduce the number of available readings incrementally, i.e. to perform partial disambiguations.

Current underspecification formalisms fall into two main classes, we could name *enumerative* approaches and *constraint-based* approaches. In the enumerative approaches, such as Hobbs&Shieber's algorithm [10] or QLF [1], the underspecified representation is a partial formula, easy to obtain from the syntactic structure of the sentence, which contains unsolved terms. Such a formula is disambiguated by means of a suitable method; in the two cited approaches, the unsolved terms are "pulled out" and "unstored", obeying to certain constraints, until no one of them appears in the formula. Depending on the order chosen to solve them, different readings are obtained. Basically, enumerative approaches

¹ We address the reader to [15], in which a possible semantics for DTS is outlined in terms of a comparison with the logics developed by [21] and [22] for dealing with branching quantification and generalized quantifiers.

generate all possible readings of a sentence in a single run of the disambiguation process, which is considered as a non-separable whole; consequently, partial disambiguations are difficult to manage (although not impossible, see [2]). On the contrary, the constraint-based methods allow one to reduce the number of available readings by adding constraints in a monotonic way. Formalisms belonging to this second approach are, for example, UDRT [20], MRS [5] and CLLS [9]. In these formalisms, the underspecified representation is a flat set of labelled formulae which contains particular variables named holes. Disambiguation is performed by assigning labels to holes. Moreover, it is possible to specify dominance constraints between holes and labels in order to reduce the number of available readings but, as stated above, these constraints can be added monotonically.

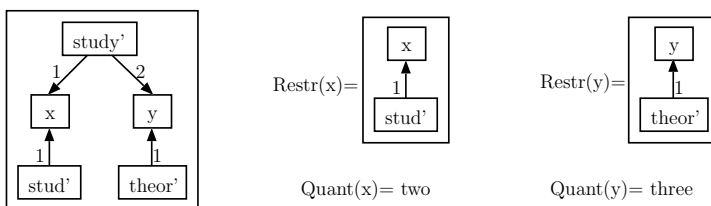
Nevertheless, enumerative approaches are basically simpler than constraint-based ones. Therefore, in the latter, the definition of the syntax-semantics interface is more difficult. [11] propose such a definition from HPSG [19] to MRS, and [11] from LTAG [12] to MRS. However, these interfaces are quite complex and do not scale up, without relevant modifications, when the linguistic coverage increases.

2 Dependency Tree Semantics

Dependency Tree Semantics (DTS) is a semantic underspecified formalism for dealing with quantifier scope ambiguity. DTS tries to keep the advantages of both enumerative and constraint-based approaches minimizing, in this way, the aforementioned trade-off between flexibility with respect to the syntactic side and flexibility with respect to the pragmatic side: it has a straightforward syntax-semantics interface with a Dependency Grammar and it allows for monotonically adding constraints to take partial disambiguations into account.

DTS specifies quantifier scope by explicitly showing the dependencies between involved (quantified) groups of entities. DTS well-formed structures are based on a simple graph G representing the predicate-arguments relations, without any quantification. The nodes of G are either predicates or discourse referents; each arc connects a predicate with a discourse referent and is labelled with the predicate argument position. With each discourse referent we associate a *quantifier* (given by a function *QUANT* from discourse referents to quantifiers) and its *restriction*, which is given by a function *RESTR* that associates a subgraph of G to each discourse referent. In (1), we show a first simple example

(1) Two students study three theorems



The representation in (1) is still ambiguous; to disambiguate, we need to specify how the quantifiers depend on each other. This is done by inserting dotted arcs between discourse referents, named *semdep* arcs². In figure 1.a and fig 1.b two fully-specified representations of sentence (1) are given. Fig.1.a shows the reading in which the quantifier ‘three’ depends on (has scope inside) the quantifier ‘two’. In figure 1.b, the arc linking x to y specifies that the two students depend on the theorems. In both interpretations, the wide-scope quantifier is linked to a new node called *Ctx* (the context).

But DTS allows for very natural representation of a third reading of sentence (1): in figure 1.c, both discourse referents are linked to the context. This is the branching quantifier (BQ) reading. The BQ reading is true only in those models in which we can find a set of two students and a set of three theorems, for which it holds that each student in the first set studies each theorem in the second one. In NL, there are many cases in which the correct truth conditions can be

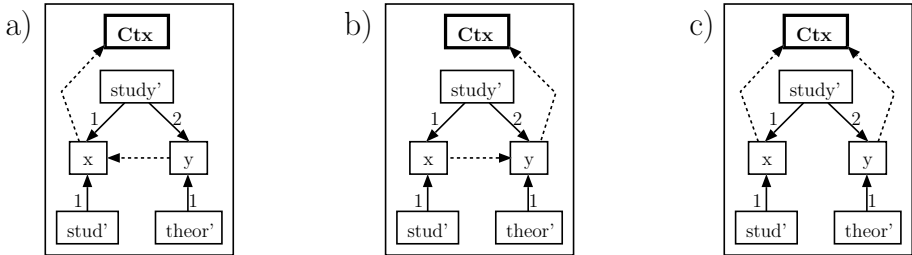


Fig. 1. The three readings of sentence (1)

handled only via a BQ reading; in fact, it is easy to add some context elements in order to force the two involved sets to be constant; for instance, in (2.1), the involved students and theorems are explicitly mentioned in two appositions, while in (2.2) the prepositional modifier *with my sister* favours an interpretation in which three persons, two friends of mine and my sister, went together to three same concerts. Finally, even if there are no explicit syntactic elements forcing a BQ reading, in many cases this is done by world knowledge; for example, in (2.3), world knowledge seems to favour the reading in which two students have seen the same three drug dealers the most salient.

- (2)
1. Two students, *John and Jack*, study three theorems: *the first three of the book*.
 2. Two friends of mine went to three concerts *with my sister*.
 3. Two students of mine have seen three drug dealers in front of the school.

² Transitive (redundant) precedences will never be shown in the figures.

2.1 Constraints on *semdep* Arcs

It is clear that not all possible configurations of *semdep* arcs are allowed; consider, for example, fig.2, which shows (in a compact form due to space constraints) two possible configurations for sentence (3)

(3) Every representative of a company saw most samples [10]

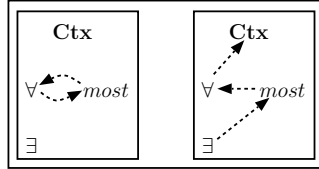


Fig. 2. Incorrect configurations for (3)

Semdep arcs denote dependencies between involved entities, therefore, in fig.2 on the left, the representatives depend on a sample and, at the same time, the samples depend on a representative. Instead, in fig.2 on the right, each representative saw a different set of samples and, for each sample he saw, he belongs to a different company. Clearly, these readings make no sense and must be disallowed. [10] prevents them by blocking certain derivations in the disambiguation process, which would lead to ill formed formulae containing free variables; the DTS's counterpart is a set of constraints on *semdep* arc insertion. However, such constraints are not only based on formal grounds: rather than avoiding syntactically ill-formed formulae, they avoid well formed ones which contain paradoxical sets of involved entities. Clearly, an analysis of the actual meaning of the formulae provides a better motivation than pure formal limitations of the logics chosen to represent this meaning, which could simply be not expressive enough to handle NL semantics. The set of constraints on *semdep* arc insertion is:

- (4) 1. The transitive closure of **SemDep** is a partial order on all discourse referents and **ctx**, with **ctx** as its maximal element.
2. Let d be a discourse referent, and let $R(d)$ be the smallest set that contains d , and for which it holds that if d' is in $R(d)$ and d'' occurs in the restriction of d' , then also $d'' \in D$. It must hold that:
 - (a) If $d_1 \in R(d)$, $d_2 \notin R(d)$, and d_1 depends on d_2 , then d depends on d_2 .
 - (b) If $d_1 \in R(d)$, $d_2 \notin R(d)$, and d_2 depends on d_1 , then d depends on d_1 .

Moreover, we add the other two following constraints to avoid redundant readings in case universal or existential quantifiers occur in the sentence.

- (5) 1. A node corresponding to a universal quantifier can only enter **Ctx** or another quantifier in its syntactic restriction.
2. No arc can enter a node corresponding to an existential quantifier.

The problem of redundant readings is not new in underspecification (see, for example, [4]); in fact, consider sentences (6.a-c) and configurations in fig.3.

- (6) a. Two students have studied all theorems.
 b. Some friends of two participants arrive.
 c. Every friend of every participant arrives.

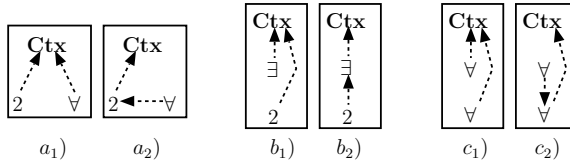


Fig. 3. Six configurations for sentences in (6)

It is easy to realize that a_1 and a_2 in fig.3 are logically equivalent: in a_2 we must choose, for each student, of a set composed *by all theorems in the domain*. Of course, this means that actually there is no choice. Therefore, a_2 conflates in a_1 (in which such a constant set is taken directly from the context) and must be disallowed to avoid redundancy. The same holds for the existential quantifier in b_1 and b_2 : the latter requires to select a set of one friend and, for each individual in this set, two potentially different theorems. Obviously, we obtain the same result in b_1 , so b_2 is redundant. Nevertheless, there is an exception to the rule about universals. It concerns an universal acting as a modifier for another quantifier in its restriction. In this case, a *semdep* arc from an universal quantifier to this last leads to a separate reading. Intuitively, this is due to the fact that the restriction sets up a new set of contexts (one for each individual satisfying the restriction). Consequently, the upper universal can refer either to the overall context or to the sub-contexts activated by its restriction. In fact, for (6.c) there are two readings, according to the intention of referring to 'all friends of any participant' (fig.3, c_1) or to 'all friends of all participants' (fig.3, c_2).

Constraints in (4) and (5) allow for six readings of (7). Fig.4 shows the underspecified representation of (7) whereas fig.5 its six readings. The actual admissibility of the readings is confirmed by linguistic intuition: for each of them it is possible to find another sentence, with the same syntactic structure and logical quantifiers of (7), in which the reading is more reasonable. In (8.a-f) we report such examples³ for readings in fig.5 (from left to right).

- (7) Two politicians spy on someone from every city. [14]
- (8) a. Two boys, John and Jack, went (together) to a show about all rockstars.
 b. Two boys, John and Jack, visited (together) a museum in every city.
 c. Two boys, John and Jack, (each) bought an atlas with all countries.
 d. At least two soldiers guard a door of every building.
 e. At least two women were sitting in a central seat of each table.
 f. Two boys, John and Jack, ate a portion of every course of the menu.

³ Note that a discourse referent can depend on more than one other discourse referent; e.g., in (8.f), we have a different portion for each pair (*boy*, *course*).

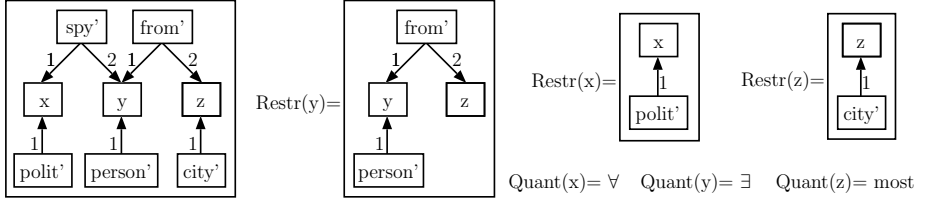


Fig. 4. Underspecified representation of (7)

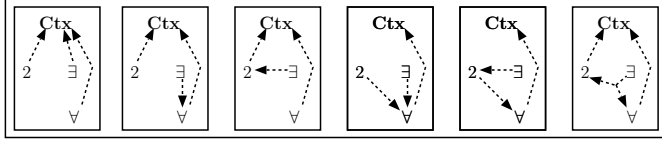


Fig. 5. The six disambiguated readings for (7)

2.2 Formalisation

In this section, formal definitions of DTS syntax and the $DG \Rightarrow DTS$ interface, where DG is a generic dependency grammar, are provided.

Syntax of DTS. A wff structure in DTS is a Scoped Dependency Graph (SDG) as defined below. As in standard semantics, the definitions below are stated in terms of three sets: a set of predicates $\mathbf{pred}$, a set of constants $\mathbf{name}$ and a set of variables D named *discourse referents* as in DRT [13]. Moreover, we write $P^1 \subseteq \mathbf{pred}$ for the unary predicates, $P^2 \subseteq \mathbf{pred}$ for the binary ones and $\iota_{\mathbf{name}} \subseteq P^1$ for those obtained by applying the ι -operator to a constant in $\mathbf{name}$ ⁴.

Definition 1. [Flat Dependency Graphs (FDG)]

A FDG G_f is a tuple $\langle N, L, A, Dom, f \rangle$ s.t.:

- N is a set of nodes $[n_1, n_2, \dots, n_k]$.
- L is a set of labels $[l_1, l_2, \dots, l_m]$; in all FDGs shown above, $L \equiv \{1, 2\}$.
- A is a set of arcs; an arc is a triple (n_s, n_d, l) , where $n_s, n_d \in N$ and $l \in L$.
- $Dom \equiv \mathbf{pred} \cup D$ is a set of domain objects.
- f is a function $f : N \mapsto Dom$, specifying the node referent, i.e. the domain object with which the node is associated⁵.

Definition 2. [Scoped Dependency Graph (SDG)]

A SDG G_s is a tuple $\langle G_f, \mathbf{ctx}, Q, \mathbf{quant}, \mathbf{restr}, \mathbf{SemDep} \rangle$ s.t.

⁴ The ι -operator has the standard semantics: if α is a constant, $\iota\alpha$ is a unary predicate which is true only for α .

⁵ In the following, when $f(n)=u$, $u \in X$ and $X \subseteq \mathbf{pred} \cup D$, we will informally say that node n is of type X , denoted by $[n]^X$. For instance, the node n is of type P^2 , i.e. $[n]^{P^2}$, if $f(n)=\mathit{love}'$ and $\mathit{love}' \in P^2$.

- $G_f = \langle N, L, A, Dom, f \rangle$ is an FDG.
- ctx is a special element called the context.
- Q is a set of quantifiers [every, most, two, ...]
- $quant$ is a total function $N_D \mapsto Q$, where $N_D \subseteq N$ are the nodes of type D .
- $restr$ is a total function assigning to each $d \in N_D$ a subgraph of G_f .
- $SemDep$ is a relation $N_D \times (N_D \cup \{\{ctx\}\})$ satisfying constraints in (4).

If $SemDep$ is not defined for each $d \in N_D$, G_s is said 'underspecified'. Note that it is not required that $SemDep$ satisfy constraints in (5).

DG $\Rightarrow$ DTS Interface. The DG $\Rightarrow$ DTS interface is a generalization of Extensible Dependency Grammar (XDG) ([6], [8]) which represents linguistic knowledge along several levels of description, called *dimensions*. DG $\Rightarrow$ DTS interface defines two of such dimensions: Predicate Argument (PA), containing an SDG G_s , and Immediate Dominance (ID), containing a Dependency Tree D_t . A Dependency Tree is defined as an FDG (def.1), with the following differences

- D_t is a Tree rather than a D.A.G.
- $Dom \equiv TV \cup PN \cup CN \cup PREP \cup DET$, i.e. Dom is the union of six well-know POS, each of which is a set of lemmata: transitive verbs, proper names, common names, prepositions and determiners⁶.
- $L \equiv \{Subj, Obj, Dcomp, Pcomp, Pmod\}$, i.e. the possible grammatical relations are subject, object, determiner and prepositional complements and prepositional modifiers.
- A satisfies the usual valency constraints; e.g. a transitive verb must be linked to either a proper name or a determiner and the label on the arc must be either *Subj* or *Obj*, etc.

Contrary to XDG, in the DG $\Rightarrow$ DTS interface the two dimensions have *different* sets of nodes; hence, we need a mapping from the nodes of the trees in *ID* into the nodes of the graphs in *PA*. This mapping is implemented by means of two functions *Sem* and *SVar* (*Lex* and *LVar* implement the inverse mapping; we omit their definition) defined as follow:

Definition 3. [*Sem and SVar functions*]

Sem is defined for each node n in D_t of type $TV \cup PN \cup CN \cup PREP$ and *SVar* for each node n in D_t of type $PN \cup DET$ according to the following constraints

- for all $[n]^{TV \cup PREP}$, $Sem(n)$ is of type P^2 .
- for all $[n]^{CN}$, $Sem(n)$ is of type P^1 .
- for all $[n]^{PN}$, $Sem(n)$ is of type *iname*.
- for all $[n]^{PN \cup DET}$, $SVar(n)$ is of type D .

Note that nodes of type PN belong to the domain of both *Sem* and *SVar*. Clearly, this is done for homogeneity of representation, as in DRT, in which a proper name yields both a discourse referent and a DRS-condition.

Having defined *Sem* and *SVar*, it is possible to state how to obtain a FDG from the Dependency Tree. This is enforced by a set of if-then rules. Some

⁶ Quantifiers are determiners, i.e. belongs to DET.

of these rules, reported in fig.6, map the mandatory syntactical realizations, i.e. the complements, into suitable predicate-argument relations. The first rule states

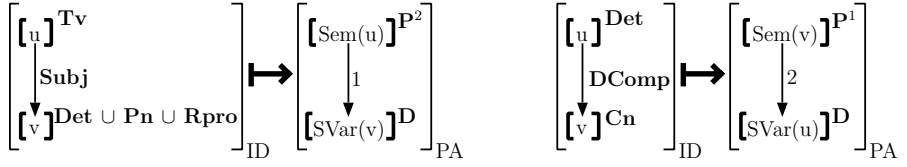


Fig. 6. DT⇒FDG Interface: Complements

that the discourse referent associated with the subject of a transitive verb must be the first argument of the predicate corresponding to this verb (a similar rule applies to objects). The second example rule specifies that the discourse referent associated with a determiner is the first argument of the predicate corresponding to the common name as complement of the determiner. Then, the first rule

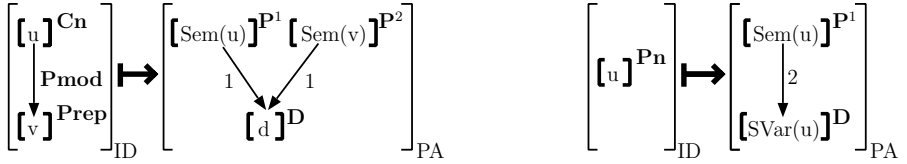


Fig. 7. DT⇒FDG Interface: Adjuncts

in fig.7 shows how an prepositional modifier is mapped, while the second one concerns the mapping of proper names. Finally, the values of the function **restr** have to be set⁷ for each node of type *D* according to the following definition

Definition 4. *[restr]*

For each $[d]^D$, let T_d be the subtree of D_t having $Lex(d)$ as root and let G_d be the subgraph of G_s obtained from T_d by means of the $DG \Rightarrow DTS$ interface. Then, **restr**(d) is the subgraph of G_d composed by all the nodes $[p]^{pred}$ linked to $[d]^D$ and all nodes $[d_1]^D$ linked to $[p]^{pred}$.

Definition 4 simply states that **restr**(d) is the subgraph of the FDG in *PA* containing all and only the nodes-predicate having d as argument and corresponding, in the Dependency Tree, to a node-lemma placed *under* $LVar(d)$ ⁸.

The $DG \Rightarrow DTS$ interface is very close both to the translation rules $DSyntR \Rightarrow SemR$ of Meaning $\Leftrightarrow$ Text Theory ([17] and [18]) and to XDG. In fact,

⁷ The function **quant**, instead, is a simple lexical function associating an element of *Q* to each node of type *D*, therefore we omit its precise formalization.

⁸ More a predicate $[p]^{name}$, when $LVar(d)$ is of type *PN* and $Sem(LVar(d))=p$.

this interface tries to keep the advantages of both theories: it is as general as MTT (XDG makes the strong assumption that the syntactic and semantic structures share the same set of nodes) but preserves the main advantage of XDG wrt MTT, i.e. an higher integration between the two structures, a property which can easily lead to interleaved approaches for the NLP (see [7]). As final examples, consider the Dependency trees in fig.8; the reader can check that the $DG \Rightarrow DTS$ Interface presented in this section allows one to obtain the SDGs in fig.1 and fig.4 respectively. In such trees, as well as in the SDG shown above, each node n is labelled with the value of $f(n)$ in order to increase readability.

Sem(study)=study'
 Sem(theorem)=theor'
 Sem(student)=stud'
 Sem(spy-on)=spy'
 Sem(politician)=polit'
 Sem(city)=city'
 Sem(one)=person'
 Sem(from)=from'

 SVar(every)=z
 SVar(some)=y
 SVar(two)=x
 SVar(three)=y

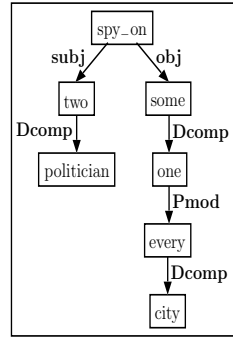
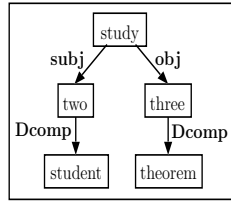


Fig. 8. Dependency Trees of (1) and (7)

3 Conclusions and Further Works

In this paper, a new underspecified formalisms, named Dependency Tree Semantics has been presented. In particular, we showed formal definitions of the DTS syntax and of the syntax-semantics interface, for compositionally building the DTS well formed structure corresponding to the syntactic (dependency) tree of a sentence. Dependency Tree Semantics encloses advantages of main proposals in Semantic Underspecification and in Semantic Interpretation of Dependency Grammars: it has a simple syntax-semantics interface as QLF, but does not merge predicate-argument relations with scope constraints, enabling partial disambiguations, as UDRT, MRS or CLLS; it has an integrated syntax-semantics interface as XDG but it is general as MTT.

However, the main contribution brought by DTS is the possibility of representing branching quantifiers readings. This allows DTS to accept readings in a far higher number than commonly accepted in other approaches. We are currently investigating the extension of DTS for handling cumulative readings. In fact, as argued in [21], cumulative readings, which received very few attention in underspecification, simply derive from another type of branching quantification. A semantics of DTS well formed structures may be found in [15].

References

1. H. Alshawi. 1992. *The Core Language Engine*. Mit Press, Cambridge, UK
2. H. Alshawi and R. Crouch. 1992. *Monotonic Semantic Interpretation* Proc. Of the 30th Annual Meeting of the ACL, Mit Press 32–39
3. H. Bunt. 2003. Underspecification in Semantic Representations: Which Technique for What Purpose?. *5th Workshop on Computational Semantics (IWCS-5)*, 37–54.
4. R.P. Chaves. 2003. Non-Redundant Scope Disambiguation in Underspecified Semantics. In *Balder ten Cate (Ed.) Proc. of the 8th ESSLLI Student Session*, 47–58.
5. A. Copestake and D. Flickinger and I.A. Sag. 1999. *Minimal Recursion Semantics. An introduction*. Manuscript, Stanford University
6. Ralph Debusmann and Denys Duchier and Alexander Koller and Marco Kuhlmann and Gert Smolka and Stefan Thater. 2004a. A Relational Syntax-Semantics Interface Based on Dependency Grammar. *Proc. 20th COLING*, Geneva
7. Ralph Debusmann and Denys Duchier and Geert-Jan M. Kruijff 2004b. Extensible Dependency Grammar: A New Methodology. *Proc. of the COLING 2004 Workshop on Recent Advances in Dependency Grammar*, Geneva
8. R. Debusmann and D. Duchier and A. Rossberg 2005. Modular Grammar Design with Typed Parametric Principles. *Proc. of FG-MOL 2005*, Edinburgh
9. M. Egg, A. Koller and J. Niehren 2001. *The Constraint Language for Lambda Structures*, Journal of Logic, Language and Information, 10:457–485.
10. J.R. Hobbs and S. Shieber 1987. An Algorithm for Generating Quantifier Scoping. *Computational Linguistics*, 13:47–63.
11. A. K. Joshi and L. Kallmeyer 2003. *Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG* Research on Language and Computation. 1:3–58
12. A.K. Joshi and Y. Schabes 1997. *Tree-Adjoining Grammars* In G. Rozenberg, A. Salomaa (eds.), Berlin, Springer Verlag 69–123
13. H. Kamp and U. Reyle. 1993. *From Discourse to Logic*. Kluwer Academic Publishers. Dordrecht.
14. R.K. Larson 1985. Quantifying into NP. *Ms. MIT*
15. L. Lesmo, L. Robaldo and J. Gerbrandy. 2006. Quantifiers in Dependency tree Semantics Proc. Inference in Computational Semantics (ICOS-5).
16. R. May and A. Bale. 2002. Inverse Linking. *Encyclopedia Entry in Syncom*, <http://kleene.sss.uci.edu/~rmay>
17. I. Mel'cuk 1988. Dependency syntax: theory and practice. *SUNY University Press*
18. I. Mel'cuk 2000. Semantics and the Lexicon in modern Linguistics. In *Proc. of the 1st Int. Conf. on Intelligent Text Processing (CICLing)*, 6–18
19. C. Pollard and I. Sag 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press
20. U. Reyle 1993. *Dealing with ambiguities by Underspecification: Construction, Representation and Deduction* Journal of Semantics, 123–179
21. G. Sher 1990. *Ways of branching quantifiers* Linguistics and Philosophy 13:393–422
22. G. Sher 1997. *Partially-ordered (branching) generalized quantifiers: a general definition*. The Journal of Philosophical Logic, 26:1–43

Mining Tolerance Regions with Model Trees

Annalisa Appice and Michelangelo Ceci

Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70126 Bari - Italy
`{appice, ceci}@di.uniba.it`

Abstract. Many problems encountered in practice involve the prediction of a continuous attribute associated with an example. This problem, known as regression, requires that samples of past experience with known continuous answers are examined and generalized in a regression model to be used in predicting future examples. Regression algorithms deeply investigated in statistics, machine learning and data mining usually lack measures to give an indication of how “good” the predictions are. Tolerance regions, i.e., a range of possible predictive values, can provide a measure of reliability for every bare prediction. In this paper, we focus on tree-based prediction models, i.e., model trees, and resort to the inductive inference to output tolerance regions in addition to bare prediction. In particular, we consider model trees mined by SMOTI (Stepwise Model Tree Induction) that is a system for data-driven stepwise construction of model trees with regression and splitting nodes and we extend the definition of trees to build tolerance regions to be associated with each leaf. Experiments evaluate validity and quality of output tolerance regions.

1 Introduction

Regression trees extend well-known decision trees to deal with a continuous goal variable Y [1]. Given a training set $D = \{(\mathbf{x}, y) \in \mathbf{X} \times Y \mid y = g(\mathbf{x})\}$, regression trees are built *top-down* by recursively partitioning a feature space $\mathbf{X}$ spanned by m independent (or predictor) variables x_i (both continuous and categorical) to mine a piecewise function f that is hopefully close to g on the domain $\mathbf{X}$. Regression trees differ from decision trees in associating continuous constant values rather than discrete classes with leaves.

Model trees generalize regression trees in the sense that they approximate g by means of a piecewise (linear) function instead of a constant value. They are mined in form of hierarchies of nodes starting with a root node (top-level node), where internal nodes (splitting nodes) are generally associated with a logical test on predictor variables, while leaves (i.e. bottom nodes in the hierarchy) are associated with (linear) functions [16,10,19,17,18,2,11,4]. A different tree structure with both regression and splitting nodes is mined by the system SMOTI [12]. Regression nodes perform straight-line regression, while splitting nodes partition the training space.

Model trees mined by SMOTI are proved to be characterized by understandability properties and valuable predictive capabilities [12]. Anyway, they lack

measures (confidence values) to give an indication of how good the predictions are. This is a common limitation of the most of regression systems, included model tree induction systems, which output the bare prediction and lead data miners to rely on the previous experience or relatively loose theoretical upper bounds on the probability of error to ensure the quality of the given prediction.

Constructing a tolerance region [6] is a relevant notion in estimating the confidence value of a continuous prediction. Although, this is an established area in statistics, no method developed in traditional statistics takes into account high-dimensional problem typical of data mining. Some solutions come from Bayesian methods and PAC theory, which give confidence values to complement bare predictions. Anyway, Bayesian methods are only applicable if the stochastic mechanism generating the data is known in every detail [13]. In contrast, PAC theory assumes data is generated by some completely unknown i.i.d. distribution, but they give non trivial-results only when data are particularly clean [13].

Algorithmic theory of randomness is exploited to provide valid confidence information [9] within a transductive inference framework [7]. In this case, the main disadvantage is the relative computational inefficiency, since for every test all computations need to be started from scratch. To overcome inefficiency limitations of transductive inference without serious loss in the confidence of the quality values, we follow the main inspiration in [14] of replacing transductive inference with inductive inference. In particular, we resort to inductive inference to construct the tolerance regions to be associated with the leaves of model trees mined with SMOTI. The idea is to randomly split training data in proper data and calibration data. The model tree is top-down mined on proper training data, while the tolerance regions are constructed on calibration data. Tolerance regions are constructed by strictly including calibration values predicted from the model tree at each leaf with an error event with probability at most δ .

The paper is organized as follows. In the next Section, we briefly describe the system SMOTI, while in Section 3 we present two methods based on inductive inference to construct tolerance regions to be associate with leaves of model trees mined by SMOTI. Experimental results to evaluate validity and quality of tolerance regions constructed in both cases are commented in Section 4. Finally, conclusions and future works are drawn.

2 Mining Model Trees with SMOTI

For the sake of self-consistency of the work, we briefly recall some characteristics of SMOTI. SMOTI learning algorithm starts with a root node t_0 that is associated with the entire training data and recursively proceeds by: i) growing the tree by performing a splitting test on the current node t and introducing the nodes t_L (left child of t) and t_R (right child of t), ii) growing the tree by performing a regression step on some continuous attributes and introducing its unique child t_R , iii) stopping the tree's growth at the current node t . Hence, model trees are mined with two types of nodes: regression nodes, which perform only straight-line regressions, and splitting nodes, which partition training data.

Splitting and regression nodes pass down observations to their children in two different ways. For a splitting node t , only a subgroup of the $N(t)$ observations in t is passed to each child (left or right). No change is made on training cases. For a regression node t , all the observations are passed down to its only child, but the values of both the dependent and independent continuous variables not included in the multiple linear model associated with t are transformed to remove the linear effect of those variables already included. Thus, descendants of a regression node will operate on a modified training set. This is done in accordance to the statistical theory of linear regression, where the incremental construction of a multiple linear model is made by removing the linear effect of introduced variables each time a new independent variable is added to the model [5]. Leaves are associated with (multiple) linear functions built by combining straight-line regressions along the path from the root to the current node. In this way SMOTI potentially solves the problem of modeling phenomena, where some variables have a global effect while others have only a local effect and prevent problems related to two-staged induction algorithms, which first build the tree structure and then associate leaves with multiple models. On the other hand, the step-wise construction provides a solution to problems of efficiency and collinearity by selecting a subset of variables. A more detailed explanation of SMOTI and a comparison with other TDIMT methods are reported in [12].

Although experimental results prove that model trees mined by SMOTI are simple and easily interpretable, and their analysis reveals interesting patterns, no indication on the reliability of performed predictions is provided. Such reliability can be estimated in form of tolerance regions.

3 Inducting Tolerance Regions in SMOTI

Tolerance regions can be efficiently induced within the Inductive Confidence Machine (ICM) framework [14] by overcoming computational inefficiency of Transductive Confidence Machine (TCM) [15,9], where the confidence of prediction is estimated by resorting to transductive inference [7]. In this paper, we extend the ICM framework to the case of tolerance regions to be predicted with a model tree. In particular, we intend to ensure the confidence of predictive patterns mined with SMOTI by providing some practical information on the prediction reliability. To this aim, the training set $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ is randomly partitioned into two sub-sets, that is, the proper set $P = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)\}$ and the calibration set $C = \{(\mathbf{x}_{l+1}, y_{l+1}), \dots, (\mathbf{x}_n, y_n)\}$ ($l \leq n$).

SMOTI-ICM mines a model tree τ from the proper set P and builds a tolerance region for each leaf of τ by considering only the calibration examples of C reaching the leaf. Formally, let N_τ^L be the set of leaves in τ , $C(t) = \{(\mathbf{x}_{t_1}, y_{t_1}), \dots, (\mathbf{x}_{t_k}, y_{t_k})\} \subseteq C$ is the sub-set of calibration examples reaching $t \in N_\tau^L$. Fixed the confidence level $1 - \delta$, SMOTI-ICM processes $C(t)$ to induce a tolerance region $TR_t(\mathbf{x})$, to be predicted for any example $\mathbf{x} \in \mathbf{X}$ reaching t .

Given an example $(\mathbf{x}, y)$ (with possible unknown value of y) that reaches t in τ , the predicted tolerance region is $TR_t(\mathbf{x})$ such that the probability that $y \in TR_t(\mathbf{x})$ is $1 - \delta$. The tolerance region depends on the bare prediction $\hat{y} = f_t(\mathbf{x}) =$

$\beta_0 + \sum_j (\beta_j \times x_j)$ that is obtained by combining all straight-line regressions along the path from the root to t with the best straight-line regression associated with t in τ [5]. The tolerance region TR_t is built according to two procedures, i.e., the *count-based procedure* and the *normal distribution based procedure*.

3.1 Count-Based Procedure

The count-based procedure is inspired by the work of [14] and operates on the list of absolute residuals. More formally, for each leaf $t \in N_\tau^L$, the bag of the absolute residuals $AR(t)$ is built as follows:

$$AR(t) = \{ar_i \mid ar_i = |y_i - \hat{y}_i| \wedge (\mathbf{x}_i, y_i) \in C(t) \wedge \hat{y}_i = f_t(\mathbf{x}_i)\} \quad (1)$$

The bag of absolute residuals is processed to obtain $AR'(t)$ that contains absolute residuals of $AR(t)$ without repetitions. $AR'(t)$ is used to build a tolerance region predicted at the leaf t by imposing that the probability that examples fall in this region is greater or equal to $(1-\delta)$. Operatively, by denoting with:

$$count_p(t) = \#\{ar_i \mid ar_i \in AR(t) \wedge ar_i > ar'_p\} \quad p = 1, \dots, \#AR'(t) \quad (2)$$

where ar'_p is the p -th value in $AR'(t)$. In this definition, $count_p(t)$ represents the number of calibration examples $(\mathbf{x}_i, y_i) \in C(t)$ whose absolute residual $|y_i - \hat{y}_i|$ is greater than the p -th absolute residual from $AR'(t)$. Hence, the tolerance region $TR_t(\mathbf{x})$ is $[\hat{y} - \alpha_\delta(t), \hat{y} + \alpha_\delta(t)]$, where $\alpha_\delta(t)$ is estimated as follows:

$$\alpha_\delta(t) = \min\{ar'_p \mid ar'_p \in AR'(t) \wedge \delta \geq \frac{count_p(t)}{\#AR'(t) + 1}\} \quad (3)$$

Intuitively, this approach allows to *locally* build the tightest tolerance region for each leaf $t \in N_\tau^L$ with respect to $C(t)$ such that the absolute frequency of calibration examples $(\mathbf{x}_i, y_i) \in C(t)$ having $y_i \in [\hat{y}_i - \alpha_\delta(t), \hat{y}_i + \alpha_\delta(t)]$ is $(1-\delta)$ at least. This is quite different from [14], where a single tolerance region is induced on the entire calibration set C according to the single rule returned by the Ridge Regression. Such difference depends on the tree structure we are dealing with that permits to estimate a different $\alpha_\delta(t)$ for each leaf t . However, inducing a tolerance region for each leaf t may pose problems when $C(t)$ contains few examples or $C(t)$ is empty ($C(t) = \emptyset$). In this case, $\alpha_\delta(t)$ cannot be reliably estimated. To avoid this problem we *globally* estimate $\alpha_\delta(t)$ by considering the entire calibration set C when $\#C(t) \leq \varphi \times \#C$ ($\varphi \in [0, 1]$ is a user-defined parameter). In such situations global estimation is preferred to local one.

3.2 Normal Distribution Based Procedure

The normal distribution based procedure resorts to the statistical interpretation of a tolerance region in form of a confidence interval, i.e., range of values around a point estimates. In statistics, confidence intervals are the most prevalent form of interval estimation to address the issue of taking a random sample from a population and computing a statistic, such as the mean, from observed data.

A confidence interval is expressed by means of a lower and upper limit for the mean, which give an indication of how much uncertainty there is in the estimate of the true mean [8]. More in general, let W be a random variable, we denote by $\hat{\theta}(w_1, \dots, w_n)$ an estimator of some unknown population parameter θ such that $\hat{\theta}(w_1, \dots, w_n)$ is computed on a set of observations $w_1, \dots, w_n$ collected for W . We assume that $g(w)$ is the known probability density function (p.d.f.) of W and determine the value α'_δ (α''_δ) such that there is a fixed probability $\delta/2$ that $w > \alpha'_\delta$ ($w < \alpha''_\delta$). Formally:

$$\delta/2 = P(w > \alpha'_\delta) = \int_{\alpha'_\delta}^{\infty} g(w)dw \quad \left(= P(w < \alpha''_\delta) = \int_{-\infty}^{\alpha''_\delta} g(w)dw \right) \quad (4)$$

Hence, the probability that $\hat{\theta} \in [\alpha'_\delta, \alpha''_\delta]$ can be computed as follows:

$$P(\alpha'_\delta < \hat{\theta} < \alpha''_\delta) = \int_{\alpha'_\delta}^{\alpha''_\delta} g(w)dw = 1 - \delta/2 - \delta/2 = 1 - \delta \quad (5)$$

This suggests the idea that the tolerance region to be predicted for any test example can be built as a confidence interval. In particular, for each leaf node $t \in N_\tau^L$, we consider the random variable $W_t = |Y - \hat{Y}| - S_{Y\hat{Y}_t}$ where $\hat{Y} = f_t(\mathbf{X})$ and $S_{Y\hat{Y}_t} = \frac{1}{\#C(t)} \sum_{(\mathbf{x}_i, y_i) \in C(t)} (y_i - \hat{y}_i)$ and assume that values of W_t observed for calibration examples reaching t are i.i.d. distributed with respect to a Normal distribution with mean μ_{θ_t} and variance $\sigma_{\theta_t}^2$ ($W_t \sim N(\mu_{\theta_t}, \sigma_{\theta_t}^2)$). Thus $g(w)$ in Equation 5 is $g(w) = \frac{1}{\sigma_{\theta_t} \sqrt{2\pi}} \exp\left(-\frac{(w - \mu_{\theta_t})^2}{2\sigma_{\theta_t}^2}\right)$. Fixed the confidence level δ , the confidence interval for the mean of W_t is $[\mu_{\theta_t} - z^* \sigma_{\theta_t}, \mu_{\theta_t} + z^* \sigma_{\theta_t}]$, where z^* represents the point on the standard normal density curve such that the probability of observing a value greater than z^* is equal to $\delta/2$ [3].

In this setting, a test example $(\mathbf{x}_i, y_i)$ (with possible unknown value of y_i) that reaches t is correctly predicted by the confidence interval when $(|y_i - \hat{y}_i| - S_{Y\hat{Y}_t}) \in [\mu_{\theta_t} - z^* \sigma_{\theta_t}, \mu_{\theta_t} + z^* \sigma_{\theta_t}]$ that is,

$$\begin{aligned} y_i \in [\hat{y}_i - (\mu_{\theta_t} + S_{Y\hat{Y}_t} + z^* \sigma_{\theta_t}), \hat{y}_i - (\mu_{\theta_t} + S_{Y\hat{Y}_t} - z^* \sigma_{\theta_t})] \\ \cup [\hat{y}_i + \mu_{\theta_t} + S_{Y\hat{Y}_t} - z^* \sigma_{\theta_t}, \hat{y}_i + \mu_{\theta_t} + S_{Y\hat{Y}_t} + z^* \sigma_{\theta_t}] \end{aligned} \quad (6)$$

When $\mu_{\theta_t} + S_{Y\hat{Y}_t} - z^* \sigma_{\theta_t} \leq 0$, that is, the mean $S_{Y\hat{Y}_t}$ of residuals is not relatively high with respect to the variance of W_t , the tolerance region for y_i is:

$$y_i \in [\hat{y}_i - (\mu_{\theta_t} + S_{Y\hat{Y}_t} + z^* \sigma_{\theta_t}), \hat{y}_i + (\mu_{\theta_t} + S_{Y\hat{Y}_t} + z^* \sigma_{\theta_t})] \quad (7)$$

The use of the random variable W_t is motivated by the fact that this allows to identify not only upper limits, but also lower limits for the residuals when this is statistically significant. This turns to have a significant play-off when no all residuals are distributed in the neighborhood of the regression line.

Similarly to the count-based procedure, some problems may occur in constructing tolerance regions when $C(t)$ contains only few calibration examples.

This is mainly due to the assumption of normal distribution for W_t that follows from the Central Limit Theorem: the random variable W_t can be assumed normally distributed when the sum of n (in the limit of large n) independent observations for W_t has a finite variance [3]. This means that the $\alpha_\delta(t)$ must be *globally* estimated by considering the entire calibration set C when $\#C(t) \leq \varphi \times \#C$.

4 An Illustrative Example

In this section we present an illustrative example that shows how the two approaches work on a simple dataset. Let us consider the proper training set:

X	1	2	3	4	5	6	7	8	9
Y	2	4	3	4	7	7	8.5	8	10

where X is the independent variable, while Y is the response variable. We are interested in building tolerance regions to be predicted at the leaves of the model tree mined by SMOTI on the proper set above. In particular, we consider the case that the tolerance regions are built by using the calibration set C_1 , that is:

X	1	2	3	4	5	6	7	8	9
Y	5	1	5	5	6	7	7	8	11

In this case, SMOTI-ICM mines a model tree with a single leaf labeled with the straight-line regression: $f(x) = 0.9667x + 1.1111$. Tolerance regions predicted at this leaf are shown in Figure 1. They are built by using either the count-based procedure or the normal distributed based procedure. More precisely, the count-based procedure builds the tolerance region $[\hat{y} - 2.92, \hat{y} + 2.92]$, while the normal distribution based procedure returns the tolerance region $[\hat{y} - 2.508, \hat{y} + 2.508]$. This suggests the idea that the normal distribution based procedure is more conservative than the count-based one, since it seems to identify tighter intervals. Moreover, this case shows that tolerance regions built from the normal distribution based procedure may collapse in a single interval area when $\mu_{\theta_t} + S_{Y\hat{Y}_t} - z^* \sigma_{\theta_t} \leq 0$. Differently, when the calibration set is C_2 :

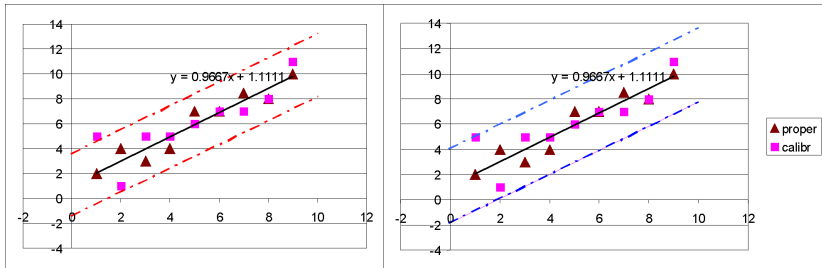


Fig. 1. Tolerance regions built on the calibration set C_1 . On the left side, the tolerance region built with the normal distribution based procedure. On the right side, the tolerance region built with the count-based procedure.

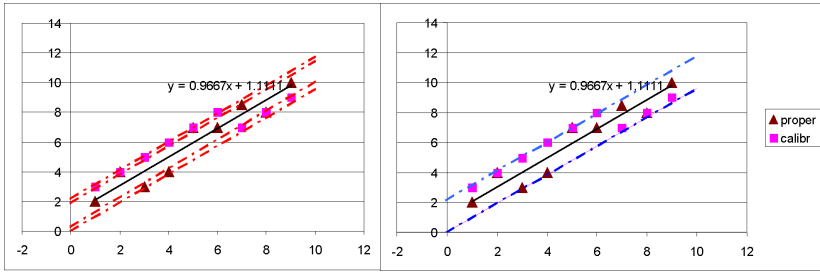


Fig. 2. Tolerance regions built on the calibration set C_2 . On the left side, the tolerance region built with the normal distribution based procedure. On the right side, the tolerance region built with the count based procedure.

X	1	2	3	4	5	6	7	8	9
Y	3	4	5	6	7	8	7	8	9

the tolerance region built with the normal distribution based procedure includes two separate interval areas (see Figure 2).

5 Experiments

Tolerance regions predicted by SMOTI-ICM are empirically evaluated on eight datasets (see Table 1) taken from either the UCI ML Repository¹ or the HTL web site², which have a continuous variable to be predicted. Experiments aim to check reliability and tightness of tolerance regions predicted by SMOTI-ICM according to both the count-based procedure (CBP) and the normal distribution based procedure (NBP). Reliability is estimated by counting how many times SMOTI-ICM fails to predict the tolerance region that contains the real value of some test example, while tightness is estimated by computing the median value of the width of the tolerance regions predicted at each leaf of the mined model tree. Tightness is estimated in terms of median value instead of mean since the median is less sensitive to noise, overfitting and outliers than the mean [14]. CBP and NBP are compared by a 10-fold cross validation of data. Reliability is estimated on the basis of the average predictive error percentage, that is:

$$AvgPredictiveError\% = \frac{1}{k} \sum_{v_i \in V} \left(\frac{1}{\#v_i} \sum_{(x_j, y_j) \in v_i} D_i(y_j, \hat{y}_j) \right), \quad (8)$$

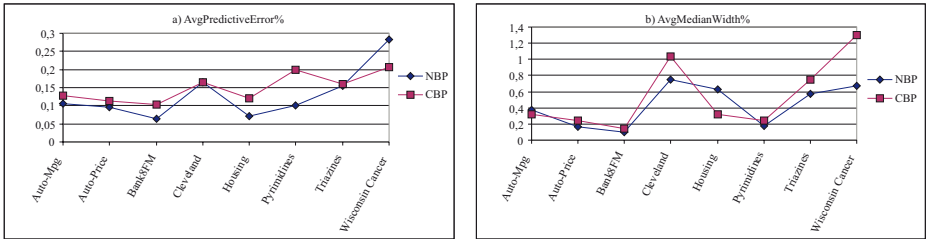
with $V = \{v_1, \dots, v_k\}$ a cross-validation partition where each v_i is a set of indices of training cases, k is the number of folds (i.e., 10), $\#v$ is the number of cases in v_i and y_j is the continuous response value for the j -th testing case from v_i . $D_i(y_j, \hat{y}_j) = 0$ if y_j belongs to the tolerance region predicted by the model tree

¹ <http://www.ics.uci.edu/~mllearn/MLRepository.html>

² [http://www.niaad.liacc.up.pt/~ltorgo/Regression/Data Sets.html](http://www.niaad.liacc.up.pt/~ltorgo/Regression/Data%20Sets.html)

Table 1. Datasets used in evaluation of tolerance regions built by SMOTI-ICM

Dataset	#Cases	#Attributes	Goal	Range of Y
Auto-Mpg	392	8	Predicting fuel consumption	[9.0,46.6]
Auto-Price	159	27	Predicting auto price	[5118,35056]
Bank8FM	4499	9	Predicting fraction of bank customers who leave the bank due to full queues	[0.0,0.8022]
Cleveland	297	14	Predicting heart disease in a patient	[0.0, 4.0]
Housing	506	14	Housing values in areas of Boston	[5.0, 50.0]
Pyrimidines	74	28	Predicting activity (QSARs) from the descriptive structural attributes	[0.1,0.9]
Triazines	74	61	Predicting structure (QSARs) from the descriptive structural attributes	[0.1, 0.9]
Wisconsin Cancer	186	33	Predicting time to recur for a breast cancer case	[1,125]

**Fig. 3.** Average a) predictive error and b) median width percentage for the tolerance regions predicted by SMOTI-ICM according to both CBP and NBP with $\delta = 0.1$

τ_i when applied to $\mathbf{x}_j$, 1 otherwise. The training set V/v_i is randomly divided into proper set P_i (70%) and calibration set C_i (30%). Each model tree τ_i is grown on P_i , while the tolerance regions at leaves of τ_i are induced on C_i .

Similarly, tightness is estimated on the basis of average percentage of the median values of the width of the tolerance regions predicted at the leaves of τ_i with respect to the range of Y in each corresponding dataset, that is:

$$AvgMedian\% = \frac{1}{k} \sum_{v_i \in V} \frac{median_{t_j \in N_{\tau_i}^L}(width(TR(t_j)))}{width(range(y))}. \quad (9)$$

Reliability of tolerance regions predicted by SMOTI-ICM is reported in Figure 3.a. Tolerance regions are built according to both CBP and NBP with $\delta = 0.1$ and $\varphi = 0.05$. In both settings, results confirm on the testing data that both procedures build tolerance regions that contain the true label y with an error event with probability close to δ . Moreover, we observe that NBP outputs tolerance regions that perform, in general, a lower error than CBP. The exception is represented by Wisconsin Cancer, where both NBP and CBP perform an error

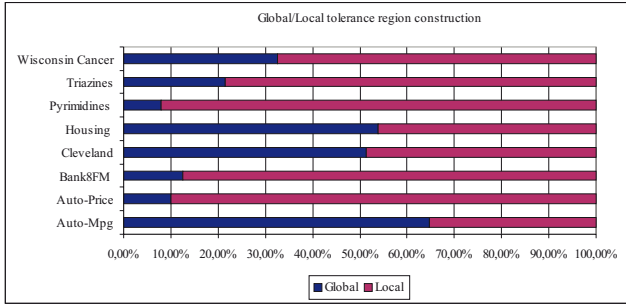


Fig. 4. Average percentage of tolerance regions globally built on the entire calibration set vs. percentage of tolerance regions locally built on sub-set of only calibration examples reaching each leaf node t

(0.28 vs. 0.20) that is significantly higher than δ . This is due to the fact that SMOTI stepwise procedure is failing in mining a piecewise function f close to the underlying function g [12]. Finally, when we compare the tightness of tolerance regions reported in Figure 3.b, we observe that low error performed by NBP is in general combined with tight tolerance regions, that is the best case.

The average percentage of tolerance regions built by using the entire calibration set (global approach) vs the average percentage of tolerance regions built by using only calibration examples reaching a leaf (local approach) are reported in Figure 4. Global construction is preferred to the local one when the percentage of calibration examples reaching the leaf node is less than the 5% (φ) of the entire calibration set. Results show that local approach is profitably preferred to the global one when enough calibration examples reach the leaf. This allows to avoid overfitting problems when building the tolerance region locally to the leaf.

6 Conclusions

In this work we resort to an inductive inference approach to construct tolerance regions as a measure of reliability of the bare prediction output by model trees mined with SMOTI. This is done by randomly splitting training data into proper data and calibration data such that the model tree structure is grown on the proper data, while the tolerance regions to be predicted at each leaf node of the tree are constructed from the calibration data. Tolerance regions are constructed according to two different procedures, named count-based procedure and normal distribution based procedure, to give an indication of how “good” the predictions are. Experimental results prove validity and quality of output tolerance regions in terms of reliability and tightness. In particular, the tolerance regions predicted by the normal distribution based procedure seem to generally correspond to the best case, that is, a lower error obtained by means of tight tolerance regions. As future work we plan to stepwise build tolerance regions to be associated with intermediate regression nodes of model trees mined by SMOTI.

Acknowledgment

This work is partial fulfillment of the research objective of COFIN-2005 project “Metodi di Data Mining Spaziali si supporto alle decisioni ambientali”.

References

1. L. Breiman, J. Friedman, R. Olshen, and J. Stone. *Classification and regression tree*. Wadsworth & Brooks, 1984.
2. P. Chaudhuri, M. C. Huang, W. Y. Loh, and R. Yao. Piecewise-polynomial regression trees. *Statistica Sinica*, 4:143–167, 1994.
3. G. Cowan. *Statistical Data Analysis*. Oxford University Press, USA, 1998.
4. A. Dobra and J. E. Gehrke. SECRET: A scalable linear regression tree algorithm. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, 2002.
5. N. R. Draper and H. Smith. *Applied regression analysis*. John Wiley & Sons, 1982.
6. D. A. S. Fraser. *Non-parametric Methods in Statistics*. New York: Wiley, 1957.
7. A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *14th Conference on Uncertainty in Artificial Intelligence*, pages 148–155, 1998.
8. G. J. Hahn and W. Q. Meeker. *Statistical Intervals : A Guide for Practitioners*. John Wiley & Sons: Series in Probability and Statistics, 1991.
9. V. V. Ilia Nourtdinov and A. Gammerman. Transductive confidence machine is universal. *Lecture Notes in Artificial Intelligence*, 2842:283 – 297, 2003.
10. A. Karalic. Linear regression in regression tree leaves. In *Proceedings of Int. School for Synthesis of Expert Knowledge, ISSEK 1992*, pages 151–163, 1992.
11. W. Y. Loh. Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 12:361–386, 2002.
12. D. Malerba, F. Esposito, M. Ceci, and A. Appice. Top down induction of model trees with regression and splitting nodes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):612–625, 2004.
13. T. Melluish, C. Saunders, I. Nourtdinov, and V. Vovk. Comparing the bayes and typicalness frameworks. In *12th European Conference on Machine Learning, ECML 2001*, volume 2167 of *LNCS*. Springer-V, 2001.
14. H. Papadopoulos, K. Proedrou, V. Vovk, and A. Gammerman. Inductive confidence machines for regression. In *13th European Conference on Machine Learning, ECML 2002*, pages 345–356, London, UK, 2002. Springer-V.
15. K. Proedrou, I. Nourtdinov, V. Vovk, and A. Gammerman. Transductive confidence machines for pattern recognition, 2001.
16. J. R. Quinlan. Learning with continuous classes. In Adams and Sterling, editors, *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, pages 343–348. World Scientific, 1992.
17. L. Torgo. Functional models for regression tree leaves. In D. Fisher, editor, *Proceedings of the Fourteenth International Conference on Machine Learning, ICML 1997*, pages 385–393, Nashville, Tennessee, 1997.
18. L. Torgo. *Inductive Learning of Tree-based Regression Models*. PhD thesis, Department of Computer Science, University of Porto, Porto, Portugal, 1999.
19. Y. Wang and I. Witten. Inducing model trees for continuous classes. In M. Van Someren and G. Widmer, editors, *Proceedings of European Conference of Machine Learning, ECML 1997*, pages 128–137, 1997.

Lazy Learning from Terminological Knowledge Bases

Claudia d'Amato and Nicola Fanizzi

Dipartimento di Informatica, Università degli Studi di Bari
Campus Universitario, Via Orabona 4, 70125 Bari, Italy
{claudia.damato, fanizzi}@di.uniba.it

Abstract. This work presents a method founded on instance-based learning algorithms for inductive (memory-based) reasoning on ABoxes. The method, which exploits a semantic dissimilarity measure between concepts and instances, can be employed both to infer class membership of instances and to predict hidden assertions that are not logically entailed from the knowledge base and need to be successively validated by humans (e.g. a knowledge engineer or a domain expert). In the experimentation, we show that the method can effectively help populating an ontology with likely assertions that could not be logically derived.

1 Introduction

Most of the research in the Semantic Web field focusses on methods based on deductive reasoning. Inductive reasoning and knowledge discovery have received less attention, with a few exceptions [1, 2]. Nevertheless, inductive reasoning may assist several important tasks that are likely to be supported by knowledge-based systems, such as clustering, classification, revision, mapping, alignment. Even retrieval may be regarded from both a deductive perspective and from an opposite one, through approximate reasoning [3].

Similarity measures for descriptions play an important role in these tasks. This has stimulated the proposal of a variety of measures for concept representations (see [4] for a survey). As pointed out in [5], most of these measures focus on the similarity of atomic concepts within hierarchies or simple ontologies, based on simple relations only. Nevertheless, the standard ontology languages are founded in Description Logics (henceforth DLs) where they borrow the language constructors. Thus, it becomes necessary to investigate the similarity of more complex descriptions expressed in DLs. It has been observed that, adopting richer representations, the structural properties have less and less impact in assessing semantic (dis)similarity.

Based on the semantics conveyed by the ABox assertions, we have proposed a dissimilarity measure between composite which is suitable for the DLs context [6]. The measure elicits the underlying semantics by querying the knowledge base for assessing the possible extension of descriptions (estimated by their *retrieval* [7]) with respect to the knowledge base [8].

In order to validate this measure, we have embedded them in algorithms which are based on the similarity between instances, namely instance-based learning. By combining an instance-based (analogical) approach with a notion of semantic distance, this

paper intends to demonstrate the applicability of inductive reasoning in this field. Therefore, the motivating application of the (dis)similarity measures might be the completion of ABoxes, through the inductive reasoning. In turn, this may trigger other related services such as learning and/or revision of faulty knowledge bases.

In particular, we have adapted a general lazy learning approach like the k -Nearest Neighbor (which reduces training phase to a mere memorization of the instances and shifts the work burden to the classification phase) to the specific Semantic Web setting. The caveat is that, in this context, the *Open World Assumption* (OWA) is generally made and multi-class problems cannot assume class disjunction by default.

The paper is organized as follows. In the next section, the representation language is briefly presented. The dissimilarity measure, recalled in Sect. 3, is exploited in a modified version of the k -NN classification procedure (Sect. 4). The experimental evaluation of the method is presented in Sect. 5. Possible developments of the method are examined in Sect. 6.

2 $\mathcal{ALC}$ Knowledge Bases

We recall the basics of $\mathcal{ALC}$ [7], a logic which adopts constructors supported by the standard ontology languages (see the DL handbook [7] for a thorough reference).

In DLs, descriptions are inductively defined starting with a set N_C of *primitive concept* names and a set N_R of *primitive roles*. Complex descriptions are built using primitive concepts and roles and the constructors showed in the following. The semantics of the descriptions is defined by an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set, the *domain* of the interpretation, and $\cdot^{\mathcal{I}}$ is the *interpretation function* that maps each $A \in N_C$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each $R \in N_R$ to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

The *top* concept $\top$ is interpreted as the whole domain $\Delta^{\mathcal{I}}$, while the *bottom* concept $\perp$ corresponds to $\emptyset$. Complex descriptions can be built in $\mathcal{ALC}$ using the following constructors. The language supports *full negation*: given any description C , denoted $\neg C$, it amounts to $\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$. The *conjunction* of concepts, denoted $C_1 \sqcap C_2$, yields an extension $C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$ and, dually, concept *disjunction*, denoted $C_1 \sqcup C_2$, yields the union $C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$. Finally, the *existential restriction*, denoted $\exists R.C$, is interpreted as the set $\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}((x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$ and the *value restriction* $\forall R.C$, has the extension $\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}((x, y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}})\}$.

The main inference is *subsumption* between concepts based on their semantics: given two descriptions C and D , C *subsumes* D , denoted by $C \sqsupseteq D$, iff for every interpretation $\mathcal{I}$ it holds that $C^{\mathcal{I}} \supseteq D^{\mathcal{I}}$. When $C \sqsupseteq D$ and $D \sqsupseteq C$ then they are equivalent, denoted with $C \equiv D$.

A *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ contains a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$. $\mathcal{T}$ is the set of definitions $C \equiv D$, meaning $C^{\mathcal{I}} = D^{\mathcal{I}}$, where C is the concept name and D is its description. $\mathcal{A}$ contains assertions on the world state, e.g. $C(a)$ and $R(a, b)$, meaning that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.

A related inference used in the following is *instance checking*, that is deciding whether an individual is an instance of a concept [9, 7]. Conversely, it may be necessary to find the concepts which an individual belongs to (*realization problem*), especially the most specific one:

Definition 2.1 (most specific concept). Given an ABox $\mathcal{A}$ and an individual a , the most specific concept of a w.r.t. $\mathcal{A}$ is the concept C , denoted $MSC_{\mathcal{A}}(a)$, such that $\mathcal{A} \models C(a)$ and for any other concept D such that $\mathcal{A} \models D(a)$, it holds that $C \sqsubseteq D$.

In case of cyclic ABoxes expressed in a DL with existential restrictions the MSC may not be expressed by a finite description [7], yet it may be often approximated.

Semantically equivalent (yet syntactically different) descriptions can be given for the same concept. Nevertheless, equivalent concepts can be reduced to a normal form by means of rewriting rules that preserve their equivalence [7]:

Definition 2.2 (normal form). A description D is in $\mathcal{ALC}$ normal form iff $D \equiv \perp$ then $D \leftarrow \perp$ or if $D \equiv \top$ then $D \leftarrow \top$ otherwise if $D = D_1 \sqcup \dots \sqcup D_n$ ($\forall i = 1, \dots, n$, $D_i \not\equiv \perp$) then

$$D_i = \bigcap_{A \in \text{prim}(D_i)} A \sqcap \bigcap_{R \in N_R} \left[\forall R. \text{val}_R(D_i) \sqcap \bigcap_{E \in \text{ex}_R(D_i)} \exists R.E \right]$$

where:

$\text{prim}(D_i)$ is the set of all (negated) primitive concepts occurring at the top level of D_i ;
 $\text{val}_R(D_i)$ is the conjunction $C_1^i \sqcap \dots \sqcap C_n^i$ in the value restriction of role R , if any (otherwise $\text{val}_R(D_i) = \top$);

$\text{ex}_R(D_i)$ is the set of concepts in the existential restrictions of the role R .

For any R , every sub-description in $\text{ex}_R(D_i)$ and $\text{val}_R(D_i)$ is in normal form.

So, for example, the concepts C and D defined as: $C \equiv A_2 \sqcap \exists R. B_1 \sqcap \forall T. (\forall Q. (A_4 \sqcap B_5)) \sqcup A_1$ and $D \equiv A_1 \sqcap B_2 \sqcap \exists R. A_3 \sqcap \exists R. B_2 \sqcap \forall S. B_3 \sqcap \forall T. (B_6 \sqcap B_4) \sqcup B_2$ are in $\mathcal{ALC}$ normal form, where A_i and B_j are all primitive concepts.

3 A Dissimilarity Measure in Description Logics

We recall the definition of dissimilarity measure for $\mathcal{ALC}$ descriptions [6]. It is a definite positive function on the set of descriptions in normal form, based on syntax and semantics of the descriptions. To define this measure, we need to assess the extensional overlap between concepts:

Definition 3.1 (overlap function). Let $\mathcal{L} = \mathcal{ALC}/\equiv$ be the set of descriptions in normal form and let $\mathcal{I}$ be the canonical interpretation of an A-Box $\mathcal{A}$. f is a function $f : \mathcal{L} \times \mathcal{L} \mapsto \mathbb{R}^+$ defined for $C, D \in \mathcal{L}$, with $C = \bigsqcup_{i=1}^n C_i$ and $D = \bigsqcup_{j=1}^m D_j$

disjunctive level:

$$f(C, D) := f_{\sqcup}(C, D) = \begin{cases} \infty & \text{if } C \equiv D \\ 0 & \text{if } C \sqcap D \equiv \perp \\ \max_{\substack{i=1, \dots, n \\ j=1, \dots, m}} f_{\sqcap}(C_i, D_j) & \text{otherwise} \end{cases}$$

conjunctive level:

$$f_{\sqcap}(C_i, D_j) := f_P(\text{prim}(C_i), \text{prim}(D_j)) + f_{\forall}(C_i, D_j) + f_{\exists}(C_i, D_j)$$

primitive concepts:

$$f_P(\text{prim}(C_i), \text{prim}(D_j)) := \frac{|P(C_i) \cup P(D_j)|}{|(P(C_i) \cup P(D_j)) \setminus (P(C_i) \cap P(D_j))|}$$

where $P(C) = \bigcap_{P \in \text{prim}(C)} P^{\mathcal{I}}$ and $f_P(\text{prim}(C_i), \text{prim}(D_j)) = \infty$ if $P(C_i) = P(D_j)$.

value restrictions:

$$f_V(C_i, D_j) := \sum_{R \in N_R} f_{\sqcup}(\text{val}_R(C_i), \text{val}_R(D_j))$$

existential restrictions:

$$f_{\exists}(C_i, D_j) := \sum_{R \in N_R} \sum_{k=1}^N \max_{p=1, \dots, M} f_{\sqcup}(C_i^k, D_j^p)$$

where $C_i^k \in \text{ex}_R(C_i)$ and $D_j^p \in \text{ex}_R(D_j)$ and we suppose w.l.o.g. that $N = |\text{ex}_R(C_i)| \geq |\text{ex}_R(D_j)| = M$, otherwise the indices N and M as well as C_i and D_j are to be exchanged in the formula above.

The function f represents a measure of the overlap between two descriptions (namely C and D) expressed in $\mathcal{ALC}$ normal form. It is defined recursively beginning from the top level of the descriptions (a disjunctive level) up to the bottom level represented by (conjunctions of) primitive concepts.

Now, it is possible to derive a dissimilarity measure based on f as follows.

Definition 3.2 (dissimilarity measure). Let $\mathcal{L}$ be the set of descriptions in $\mathcal{ALC}$ normal form and let f be the overlap function. The dissimilarity measure d is a function $d : \mathcal{L} \times \mathcal{L} \mapsto [0, 1]$ such that, $\forall C, D \in \mathcal{L}$, $C = \bigsqcup_{i=1}^n C_i$ and $D = \bigsqcup_{j=1}^m D_j$:

$$d(C, D) := \begin{cases} 1 & \text{if } f(C, D) = 0 \\ 0 & \text{if } f(C, D) = \infty \\ 1/f(C, D) & \text{otherwise} \end{cases}$$

The function d measures the level of dissimilarity between two concepts, say C and D , using the function f that expresses the amount of overlap between the two concepts. Particularly, if $f(C, D) = 0$, i.e. there is no overlap between the considered concepts, then d must indicate that the two concepts are totally different, indeed $d(C, D) = 1$, the maximum value of its range. If $f(C, D) = \infty$ this means that the two concepts are totally overlapped and consequently $d(C, D) = 0$ that means that the two concept are indistinguishable, indeed d assumes the minimum value of its range. If the considered concepts have a partial overlap then their dissimilarity is inversely proportional to their overlap, since in this case $f(C, D) > 1$ and consequently $0 < d(C, D) < 1$.

The measure baseline (i.e. the counts on the extensions of primitive concepts) depends on the KB semantics, as conveyed by the ABox assertions. This is in line with the ideas in [8, 5], where semantics is elicited as a probability distribution over the domain.

The notion of *Most Specific Concept* MSC is commonly exploited for lifting individuals to the concept level. On performing experiments related to a similarity measure exclusively based on concept extensions, we noticed that, resorting to the MSC, for adapting that measure to the individual to concept case, just falls short: indeed the MSCs may be too specific and unable to include other (similar) individuals in their extensions. By comparing descriptions reduced to the normal form we have given a more structural definition of dissimilarity. However, since MSCs are computed from the same ABox assertions, reflecting the current knowledge state, this guarantees that structurally similar representations will be obtained for semantically similar concepts.

Let us recall that, given the ABox, it is possible to calculate the most specific concept of an individual a w.r.t. the ABox, $MSC(a)$ (see Def. 2.1) or at least its approximation $MSC^k(a)$ up to a certain description depth k . In the following we suppose to have fixed this k to the depth of the ABox, as shown in [10]. In some cases these are equivalent concepts but in general we have that $MSC^k(a) \sqsupseteq MSC(a)$.

Given two individuals a and b in the ABox, we consider $MSC^k(a)$ and $MSC^k(b)$ (supposed in normal form). Now, in order to assess the dissimilarity between the individuals, the d measure can be applied to these descriptions as follows:

$$d(a, b) \leftarrow d(MSC^k(a), MSC^k(b))$$

Analogously, the dissimilarity between an individual a and a concept C is:

$$\forall a : d(a, C) \leftarrow d(MSC^k(a), C)$$

These cases may turn out to be handy in several tasks, namely both in inductive reasoning (construction, repairing of knowledge bases) and in information retrieval.

We proved [6] that d is really a dissimilarity measure according to the definition:

Proposition 3.1. *d is a dissimilarity measure for $\mathcal{ALC}/\equiv$.*

The computational complexity of the dissimilarity measure d depends on the complexity of f that, in turn, relies on reasoning services, namely subsumption and instance-checking. Hence the complexity depends on the complexity of these inferences too.

According to the definition, it is necessary to assess the complexity of f_P , $f_\forall$, $f_\exists$. As for f_P , it depends on the instance checks made for determining the concept extensions. The computations of $f_\forall$ and $f_\exists$ apply recursively the definition of $f_\sqcup$ on less complex descriptions. Particularly, $|N_R|$ calls of $f_\sqcup$ are needed for computing $f_\forall$ while the calls to $f_\sqcup$ needed for $f_\exists$ are $|N_R| \cdot N \cdot M$, where $N = |\text{ex}_R(C_i)|$ and $M = |\text{ex}_R(D_j)|$ as in Def. 3.1. These considerations show that the complexity of the computation of d strongly depends on the complexity of instance-checking for $\mathcal{ALC}$, which is P-space [9]. Obviously when computing the (dis)similarity measure for cases involving individuals, the cost of computing MSCs (approximations) has to be added.

Nevertheless, in practical applications, these computations may be efficiently carried out exploiting the statistics that are maintained by the DBMSs query optimizers. Besides, the counts that are necessary for computing the concept extensions could be estimated by means of the probability distribution over the domain.

4 k -Nearest Neighbor for Description Logics

We briefly review the basics of the k -Nearest Neighbor method (k -NN) and propose how to exploit their lazy classification procedure for inductive reasoning. In this lazy approach to learning, a notion of distance (or dissimilarity) for the instance space is employed to classify a new instance as follows.

Let x_q be the instance that we want to classify. Using the distance, the set of k nearest pre-classified instances is selected. The objective is to learn a discrete-valued target function $h : IS \mapsto V$ from a space of instances IS to a set of values $V = \{v_1, \dots, v_s\}$. In its simplest setting, the k -NN algorithm approximates h for new instances x_q on the ground of the value that h assumes in the neighborhood of x_q , i.e. the k closest instances to the new instance in terms of a dissimilarity measure. Precisely, it is assigned

according to the value which is *voted* by the majority of instances in the neighborhood. Formally, this can be written: $\hat{h}(x_q) \leftarrow \operatorname{argmax}_{v \in V} \sum_{i=1}^k \delta(v, h(x_i))$ where δ is a function that returns 1 in case of matching arguments and 0 otherwise. The problem with this simple formulation is that this setting does not take into account the similarity among instances, except when selecting the instances to be included in the neighborhood. Therefore a modified setting is generally adopted, weighting the vote on the grounds of the similarity of the instance:

$$\hat{h}(x_q) \leftarrow \operatorname{argmax}_{v \in V} \sum_{i=1}^k w_i \delta(v, h(x_i)) \quad (1)$$

where, usually, given a distance measure d , $w_i = 1/d(x_i, x_q)$ or $w_i = 1/d(x_i, x_q)^2$. Note that the hypothesized function $\hat{h}$ is defined only extensionally, that is the k -NN method does not return an intensional SURFACE-WATER-MODEL for classification (a function or a concept definition), it merely gives an answer for new query instances to be classified, employing the procedure above.

Usually this method is employed to classify vectors of (discrete or numeric) features in some n -dimensional instance space (e.g. often $IS = \mathbb{R}^n$). Let us now turn to adapt the method to the more complex context of DLs descriptions. Preliminarily, it should be observed that a strong assumption of this setting is that it can be employed to assign a value (e.g. a class) to a query instance among a set of values which can be regarded as a set of pairwise disjoint concepts/classes. This is an assumption that cannot be always valid. In this case, indeed, an individual could be an instance of more than one concept.

Let us consider a new value set $V = \{C_1, \dots, C_s\}$, of not necessarily pairwise disjoint classes (i.e. concepts) C_j ($1 \leq j \leq s$) that may be assigned to a query instance x_q . If the classes were disjoint as in the standard setting, the decision procedure defining the hypothesis function is the same as in Eq. (1), with the query instance assigned the *single* class of the majority of instances in the neighborhood.

In the general case, when we do not know about the disjointness of the classes (unless explicitly stated in the TBox), we can adopt another answering procedure, decomposing the multi-class problem into smaller binary classification problems (one per class). Therefore, we shall employ a simple binary value set ($V = \{-1, +1\}$). Then, for each class/concept, a hypothesis $\hat{h}_j$ is computed, iteratively:

$$\hat{h}_j(x_q) \leftarrow \operatorname{argmax}_{v \in V} \sum_{i=1}^k \frac{\delta(v, h_j(x_i))}{d(x_q, x_i)^2} \quad \forall j \in \{1, \dots, s\} \quad (2)$$

where each function h_j ($1 \leq j \leq s$), simply indicates the occurrence (+1) or absence (-1) of the corresponding assertion in the ABox: $C_j(x_i) \in \mathcal{A}$. Alternately¹, h_j may return +1 when $C_j(x_i)$ can be inferred from the knowledge base $\mathcal{K}$, and -1 otherwise.

The problem with non-explicitly disjoint classes/concepts is also related to the *Closed World Assumption* usually made in the context of KDD. That is the reason for adapting the standard setting to cope both with the case of generally non-disjoint classes and with the OWA which is commonly made in the Semantic Web context.

¹ For the sake of simplicity and efficiency, this case will not be considered in the following.

To deal with the OWA, the absence of information on whether a certain instance x belongs to the extension of concept C_j should not be interpreted negatively, as shown before. Rather, it should count as neutral information. Thus, one can still adopt the decision procedure in Eq. (2), however another value set has to be adopted for the h_j 's, namely $V = \{-1, 0, +1\}$, where the three values denote, respectively, occurrence, absence and occurrence of the opposite assertion. Formally:

$$h_j(x) = \begin{cases} +1 & C_j(x) \in \mathcal{A} \\ 0 & C_j(x) \notin \mathcal{A} \\ -1 & \neg C_j(x) \in \mathcal{A} \end{cases}$$

Again, a more complex procedure may be devised by simply substituting the notion of occurrence (absence) of assertions in (from) the ABox with the one of derivability (denoted with $\vdash$) from the whole knowledge base, i.e. $\mathcal{K} \vdash C_j(x)$, $\mathcal{K} \not\vdash C_j(x)$ and $\mathcal{K} \vdash \neg C_j(x)$, respectively. Although this may help reaching the precision of deductive reasoning, it is also much more computationally expensive, since the simple lookup in the ABox must be replaced with instance checking.

5 Experimentation

In order to assess the appropriateness and validity of the proposed method with the dissimilarity measure, we have applied it to the instance classification problem, using three (very different) ontologies: the FSM ontology and the SURFACE-WATER-MODEL ontology from the Protégé library², and our FAMILY ontology.

FSM is an ontology describing finite state machines. It is made up of 20 concepts (both primitives and defined), some of them are declared to be disjoint, 10 object properties, 7 datatype properties, 37 distinct individual names. About half of the individuals are instance of a single class and are not involved in any property.

SURFACE-WATER-MODEL is an ontology describing surface water and water quality models. It is based on the Surface-water Models Information Clearinghouse (SMIC) of the USGS. Namely, it is an ontology of numerical models for surface water flow and water quality simulation. The application domain of these models comprises lakes, oceans, estuaries etc.. These models are classified based on their availability, application domain, dimensions, partial differential equation solver, and characteristics types. It is made up of 19 concepts (both primitives and defined) without any specification about disjointness, 9 object properties, 115 distinct individual names; each of them is an instance of a single class and only some of them are involved in object properties.

FAMILY is an ontology describing KINSHIP relationships. It is made up of 14 concepts (both primitives and defined), some of them are declared to be disjoint, 5 object properties, 39 distinct individual names. Most of the individuals are instances of more than one concept, and are involved in more than one role. This ontology has been written to have a more complex instance graph than in the other cases. Indeed, in the other ontologies only some concepts are employed in the assertions (the others are defined only intensionally), while in the FAMILY ontology every concept has at least one instance asserted. The same happens for the role assertions; particularly, there are some

² <http://protege.stanford.edu/plugins/owl/owl-library/>

Table 1. Average results of the trials

	Predictive Accuracy	Omission Error	Induction Rate	Commission Error
FSM	100	0	31	0
S.-W.-M.	100	0	0	0
FAMILY	44.25	55.75	14	0

situations where roles assertions constitute a chain from an individual to another one, by means of other intermediate roles.

The classification method was applied to each ontology; namely, for every ontology, all individuals are checked as being instances of the concepts in the ontology. Specifically, for each individual in the ontology the MSC is computed and enlisted in the set of training examples. Each example is classified applying the adapted k -NN method presented in the previous section, adopting a leave-one-out cross validation procedure.

The experimental evaluation parameters are the following: *predictive accuracy*, *omission error rate*, *commission error rate*, *induction rate*. Namely, predictive accuracy measures the number of correctly classified individuals with respect to overall number of individuals. The omission error measures the amount of unlabeled individuals with respect to a concept while it was to be classified as an instance of that concept. The commission error measures the amount of individuals labelled as instances of the negation of the target concept, while they belong to that concept or vice-versa. The induction rate measures the amount of individuals that were found to belong to a concept or its negation, while this information is not derivable from the knowledge base.

By looking at Tab. 1 reporting the experimental outcomes, it is important to note that, for every ontology, the commission error was null. This means that the classifier has never made critical mistakes because no individual has been deemed as an instance of a concept while really it is an instance an disjoint class.

In particular, for the SURFACE-WATER-MODEL ontology (see Tab. 1), we note the prediction accuracy is equal to 100% and omission error and induction rate are null. This means that for the SURFACE-WATER-MODEL ontology our classifier always assigns individuals to the correct concepts but it is also never capable to induce new knowledge (indeed the induction rate is null). This is due to the fact that individuals in this ontology are all instances of the same concepts and roles, so computing their MSC, these are all very similar and so the amount of information they convey is very low.

For the same reasons, also for the FSM ontology (see Tab. 1), we have a maximal predictive accuracy. However, differently from the previous ontology, the induction rate for the FSM ontology is different, namely it is not null. Since the induction rate represents assertions that are not logically deducible from the ontology and was induced by the classifier, this figures would be positive if this knowledge is correct. Particularly, in this case the increase of the induction rate has been due to the presence of some concepts that are declared to be mutually disjoint.

Results are different for the case of the FAMILY ontology (see Tab. 2), where the predictive accuracy is lower and there have been some omission errors. This may be due to the fact that instances are more irregularly *spread* over the classes, that is they are instances of different concepts, which are sometimes disjoint. Specifically, there is a concentration of instances of the concepts Human, Child and GrandChild. Hence the

Table 2. Outcomes of the trials with the FAMILY ontologies

	Predictive Accuracy	Omission Error	Induction Rate	Commission Error
Female	64	36	7.69	0
Woman	64	36	7.69	0
Mother	0	100	5.12	0
Male	12.5	87.5	23	0
Man	12.5	87.5	23	0
Father	0	100	23	0
Human	100	0	2.56	0
Child	100	0	25.64	0
Sibling	62.5	37.5	41	0
Parent	29	71	2.56	0
Grandparent	75	25	0	0
Grandchild	100	0	36	0
Cousin	0	100	0	0
UncleAunt	0	100	14	0
average	44.25	55.75	14	0

MSC approximations that were computed are very different, which reduces the possibility of matching significantly similar MSCs. This is a known drawback of the Nearest-Neighbor methods. Nevertheless our algorithm does not make any commission error and it is able to infer new knowledge.

In general it is possible to assert that presented algorithm can infer new knowledge that a deductive reasoner is not able to discover. However to use it in an effective way, it would be applied to rich ontologies, characterized by a lot of concept and role instances asserted in a spread way. This method could be used to help a knowledge engineering in populating the A-Box. Indeed after a partial instantiation of the A-Box of an ontology, a classification process could be applied, hence the results of the classification can be checked by the knowledge engineering rather than instantiating the A-Box by hand.

6 Conclusions and Future Work

In this work we have tested a composite concept similarity measure on a method for inductive inference on DLs ABoxes, that can be exploited for predicting/suggesting missing information about individuals. Such information may be decisive for enabling a series of inductive tasks such as classification, case-based reasoning, clustering, etc.. The proposed method is able to induce new assertions, in addition to the knowledge already derivable by means of a reasoner. Then it seems to be able to enhance the standard instance-based classification. Particularly, an increase in prediction accuracy was observed when the instances are homogeneously spread.

The presented measure can be refined introducing a weighting factor for decreasing the impact of the dissimilarity between nested sub-concepts in the descriptions on the determination of the overall value. Another natural extension may concern the definition of dissimilarity measures in more expressive DLs languages. For example, a normal form for $\mathcal{ALCN}$ can be obtained based on those for $\mathcal{ALN}$ and $\mathcal{ALCN}$. Then, by exploiting the notion of existential mappings, the measure presented in this paper may be extended to the richer DL.

Besides, as mentioned, this method could be extended with different (yet still computationally tractable) answering procedures grounded on statistical inference (non-

parametric tests based on ranked distances), in order to accept answers as correct with a high degree of confidence. Furthermore, the k -NN method in its classical form is particularly suitable for the automated induction of missing values for (scalar or numeric) datatype properties of an individual as an estimate derived from the values of the datatypes for the surrounding individuals.

References

- [1] Lisi, F.A.: Principles of inductive reasoning on the semantic web: A framework for learning in $\mathcal{AL}$ -log. In Fages, F., Soliman, S., eds.: Principles and Practice of Semantic Web Reasoning, Third International Workshop, PPSWR2005. Volume 3703 of LNCS., Springer (2005)
- [2] Sheth, A., Ramakrishnan, C., Thomas, C.: Semantics for the semantic web: The implicit, the formal and the powerful. *Int. Journal on Semantic Web and Information Systems* **1** (2005) 1–18
- [3] Hitzler, P., Vrandečić, D.: Resolution-based approximate reasoning for OWL DL. In Gil, Y., Motta, E., Benjamins, V., Musen, M.A., eds.: Proceedings of the 4th International Semantic Web Conference, ISWC2005. Number 3279 in LNCS, Springer (2005) 383–397
- [4] Rodríguez, M.: Assessing semantic similarity between spatial entity classes. PhD thesis, University of Maine (1997)
- [5] Borgida, A., Walsh, T., Hirsh, H.: Towards measuring similarity in description logics. In: Working Notes of the International Description Logics Workshop. CEUR Workshop Proceedings, Edinburgh, UK (2005)
- [6] d’Amato, C., Fanizzi, N., Esposito, F.: A dissimilarity measure for concept descriptions in expressive ontology languages. In Alani, H., Brewster, C., Noy, N., Sleeman, D., eds.: Proceedings of the KCAP2005 Ontology Management Workshop, Banff, Canada (2005)
- [7] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: *The Description Logic Handbook*. Cambridge University Press (2003)
- [8] Bacchus, F.: Lp, a logic for representing and reasoning with statistical knowledge. *Computational Intelligence* **6** (1990) 209–231
- [9] Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: Deduction in concept languages: From subsumption to instance checking. *Journal of Logic and Computation* **4** (1994) 423–452
- [10] Mantay, T.: Commonality-based ABox retrieval. Technical Report FBI-HH-M-291/2000, Dept. of Computer Science, University of Hamburg, Germany (2000)

Diagnosis of Incipient Fault of Power Transformers Using SVM with Clonal Selection Algorithms Optimization

Tsair-Fwu Lee^{1,2}, Ming-Yuan Cho¹, Chin-Shiuh Shieh¹, Hong-Jen Lee¹,
and Fu-Min Fang^{2,*}

¹ National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan 807, ROC

² Chang Gung Memorial Hospital-Kaohsiung Medical Center, Chang Gung University
College of Medicine, Kaohsiung, Taiwan, ROC
fang2569@adm.cgmh.org.tw

Abstract. In this study we explore the feasibility of applying Artificial Neural Networks (ANN) and Support Vector Machines (SVM) to the prediction of incipient power transformer faults. A clonal selection algorithm (CSA) is introduced for the first time in the literature to select optimal input features and RBF kernel parameters. CSA is shown to be capable of improving the speed and accuracy of classification systems by removing redundant and potentially confusing input features, and of optimizing the kernel parameters simultaneously. Simulation results on practice data demonstrate the effectiveness and high efficiency of the proposed approach.

Keywords: Diagnosis, clonal selection algorithm, optimization, power transformer.

1 Introduction

Power transformers deliver electrical energy from generation system to customers, are vital to power systems. They need to be carefully and constantly monitored, and fault diagnosis must be carried out on time so that alarms can be given as early as possible. The most popular method for detecting incipient faults is based on dissolved gas analysis (DGA) [1], which detects degradations in the transformer oil and insulating materials prior to transformer failure. According to the IEC 599/IEEE C57 [2], these degradations produce fault-related gases such as hydrogen (H_2), oxygen (O_2), nitrogen (N_2), methane (CH_4), ethylene (C_2H_4), acetylene (C_2H_2), ethane (C_2H_6), carbon dioxide (CO_2), and carbon monoxide (CO), all of which can be detected in a DGA test. The existence of these gases means that corona or partial discharge, thermal heating, and arcing may be occurring [3]. The associated energy dissipation is least in corona or partial discharge, medium in thermal heating, and highest in arcing [4].

Various fault diagnosis techniques have been developed based on DGA data, including the conventional key gas method, the ratio-based method [5], and more

* Corresponding author.

recently artificial intelligence (AI) methods such as expert systems (ES) [6], fuzzy logic (FL) [7] and artificial neural networks (ANN) [8]. Some combinations of these methods are particularly promising [9]. The principle shortcoming of the key gas and ratio methods is that their fault diagnosis may vary from utility to utility, so no general mathematical formulation can be utilized. In the ES and FL approaches, information such as transformer size, oil volume, gassing rates, and past diagnosis results is usually needed to establish DGA standards for decision-making. Their intrinsic shortcomings are therefore the difficulty of acquiring such knowledge and maintaining an updated database. One solution is to optimize these methods through an evolutionary algorithm [10]. Novel, enhanced methods of evolution computation have been proposed to this end [11].

A traditional ANN method overcomes the shortcomings of expert systems by acquiring experience directly from the training data. Its weaknesses, however, include the need for a large number of controlling parameters, the difficulty of obtaining a stable solution, and the danger of over-fitting. To overcome these drawbacks a new neural network has been proposed for incipient fault diagnosis based on extension theory [12]. As the ANN, ES and FL approaches all have their advantages and disadvantages, hybrid approaches that combine ES, FL, ANN, and evolutionary algorithms (EA) into a whole are now being considered [13]. The SVM algorithm originally developed by Vapnik and Cortes [14] has emerged as a powerful tool for the analysis of such data. SVM has been used in many applications, such as face recognition [15], time series forecasting [16], fault detection [8], and the modeling of nonlinear dynamic systems [17].

This Paper proposes a new method, based on a clonal selection algorithm (CSA), which can improve the ability of Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs) to diagnose power transformer faults.

SVMs were first proposed by Vapnik of Lucent Technology's Bell Laboratories [18]. This idea combines the structural risk minimization (SRM) principle with statistical machine learning theory (SLT), carrying out risk minimization on a nested set of separated hyperplanes. SVM has been highly successful in solving classification and functional regression problems for a wide range of applications [19, 20]. It is especially powerful for nonlinear problems with high dimension but small sample size. It provides a unique solution and is strongly regularized, so is particularly appropriate for ill-posed problems.

2 Problem Description

Any DGA approach is intrinsic imprecise in typical, however, and requires experience in order to obtain reasonable results. Some unidentified cases occurred in IEC 599, for example, when ratio codes calculated from actual DGA results did not correspond to any of the codes associated with a characteristic fault. One must then check publication 60599, which contains an in-depth description of the five main types of faults usually found in electrical equipment in service. Publication 60599 classifies faults according to the main types that can be reliably identified by visual inspection of the equipment, after the fault has occurred in service [20]:

1. Partial discharges (PD) of the cold plasma (corona) type, with possible X-wax formation and spark-induced, small, carbonized punctures in the paper;
2. Low-energy discharges (D1), as evidenced by larger punctures or tracking in the paper or carbon particles in the oil;
3. High energy discharges (D2) with power follow-through, as evidenced by extensive carbonization, metal fusion, and possible tripping of the equipment;
4. Thermal faults below 300 °C if the paper has turned brownish (T1), and above 300 °C if the paper has carbonized (T2);
5. Thermal faults above 700 °C (T3), as evidenced by oil carbonization, metal discoloration, or fusion.

The number of characteristic faults has thus been reduced from eight in the previous IEC Publication 599, to five in the new IEC Publication 60599 (not including the normal condition). The three basic gas ratios described in IEC 599 (C_2H_2/C_2H_4 , CH_4/H_2 , and C_2H_4/C_2H_6) are also used in IEC 60599 to identify the characteristic faults. The ratio limits have also been made more precise, in order to reduce the number of unidentified cases from around 30% in IEC 599 to nearly 0% in IEC 60599.

In this study we propose a multi-layer SVM, with a new feature and RBF kernel parameter selection method, to predict incipient faults in power transformers. Our encoding technique is based on a clonal selection algorithm, which improves the accuracy of the classification by removing redundant and confusing input features. Simulations on real data demonstrate the effectiveness of our approach.

3 Proposed Fault Diagnosis Method

In many practical classification tasks we encounter problems with a high-dimensional input space, where we want to find the combination of original input features that contributes most to the task. The overall structure and main components of the proposed method are depicted in Fig.1. CSA performs an optimal search on the features and SVM classification parameters. The SVM classifier takes existing Vapnik-Chervonenkis (VC) theoretical bounds on the SVM generalization error instead of performing cross-validation [21]. This method is computationally much faster than k -fold cross-validation, since we only have to retrain once on each feature subset. In practical situations the VC error bounds generally have a higher bias and lower variance than cross-validation, and thus also reduce overfitting in the wrapper algorithm [21, 22]. Feature selection is performed in an evolutionary manner by clonal selection algorithm encoding, which also allows us to optimize the SVM parameters. This method is similar to the regularization of parameters in parallel.

As it is well known, SVMs are trained through the following optimization procedure:

$$\max_{\alpha} W^2(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad (1)$$

$$\text{subject to } 0 \leq \alpha_i \text{ for all } i = 1, \dots, n \text{ and } \sum_{i=1}^n \alpha_i y_i = 0,$$

where $k(\cdot)$ is called the kernel function [18]. The kernel is a scalar function of two vectors, x_i and x_j , which are both members of the feature space Φ , that is $k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$. Any function that meets Mercer's condition [2] can be used as the kernel

function. In this work, all SVMs use the radial basis function (RBF) as their kernel. The three kernel parameters C , ξ , and σ of an SVM model are important to the accuracy of classification. C is used to weight the penalizing relaxation variable ξ , and σ denotes the width of the RBF kernel. In the case of an SVM model, the CSA technique is exploited to optimize these kernel parameters in addition to other features.

CSA creates an improved diagnosis classification, consisting of features and parameters determined through an iterative process of selection, maturation and cloning, hypermutation, and reselection. The three free parameters C , ξ , and σ are encoded in a decimal format and represented by an antigen. The optimal combination of these parameters is recognized by its affinity with the antibody. According to immune network principles, once a new antigen arises in the body the existing network cells (i.e., antibodies) will recognize it. The successful recognition of an antigen will activate the networks, leading to proliferation of the corresponding antibodies.

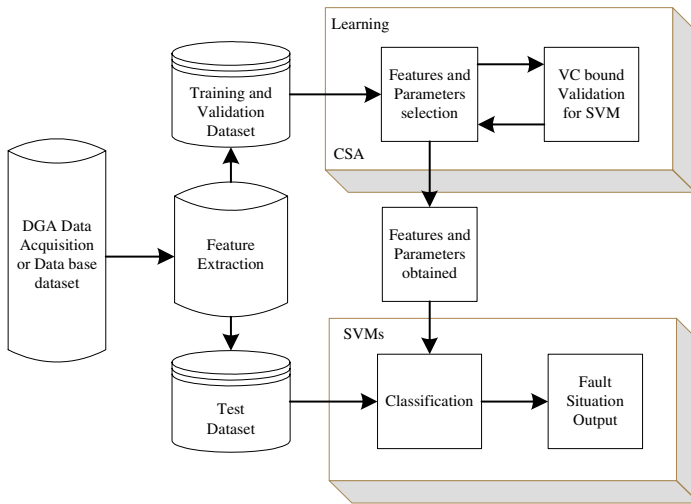


Fig. 1. Overall structure of proposed method

An alternative approach is detailed in the block diagram of our CSA-based optimization algorithm (Fig. 2) [23]. The corresponding steps are explained as follows (each sequence of six steps produces one cell generation):

1. Initialize the antibody repertoire (P_i), including the subset of memory cells (M).
2. Evaluate the fitness function for all individuals in population P . Here 'fitness' refers to the affinity measure. (Where $P = P_r + M$.)
3. Select the best candidates (P_r) from population P_i according to their affinity with the antigen.
4. Clone the best antibodies P_r into a temporary pool (C). The number of clones for each candidate is an increasing function of the antigen affinity;
5. Generate a mutated antibody pool (C'). The hypermutation (mutation with high rate) rate of each individual is also proportional to its affinity with the antigen.

6. Evaluate all the antibodies in C^l , and replace N_d antibodies with novel ones (diversity introduction). The cells with lower affinity have a higher probability of being replaced. Eliminate antibodies that are similar to those in C , and update C^l .

In this paper, we demonstrate the CSA's efficiency in selecting features and the optimal classification parameters (C , ξ , and σ) of an SVM with an RBF kernel. Simulation results demonstrate that a CSA-based approach can acquire optimal parameters and features for any given SVM classifier design criteria.

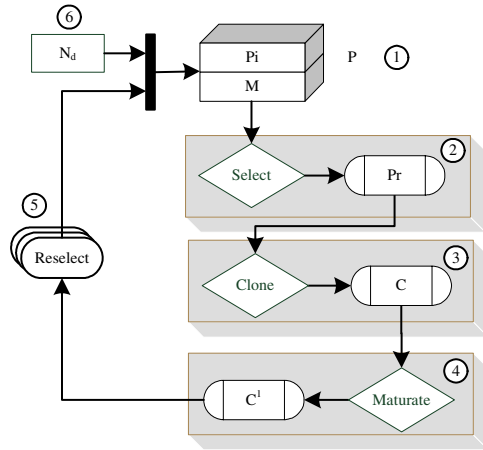


Fig. 2. Block diagram of the clonal selection algorithm

There is no superior kernel function for machine learning generally, and the performance of a kernel function rather depends on the specific application. The parameters of the kernel function also play an important role in representing the structure of a sample space. The optimal parameter set that maximizes the classification performance can be selected by a machine learning method. In the learning phase, the structure of the sample space is learned by the kernel function, and acquired knowledge of the sample space is contained in a set of parameters. The optimal set of features should also be chosen in the learning phase. In our method, the CSA technique is exploited to obtain an optimal set of features as well as the optimal parameters for the kernel function. Simulating the process of immune system adaptation, the CSA creates an improved diagnosis classification (i.e., a set of features and parameters) through an iterative process of reproduction, evaluation, maturation, clone, and reselection [24]. At the end of learning stage, the CSA-based SVM consists of a set of features and parameters for its kernel function. The CSA-based SVM obtained after the learning phase can be used to classify new DGA pattern samples in the classification phase.

Any DGA approach is inherently imprecise, however, and requires additional techniques to obtain good results. Hence, in this section we shall apply the proposed CSA-based SVM optimization algorithm to the power transformer DGA test. Most currently available software cannot effectively and automatically select features and parameters. For a SVM, this is a very complex, non-linear, non-quadratic global

optimization problem. In general the largest margin is chosen by minimizing the number of support vectors. Lee and Lin [25] make full use of their training samples, adopting an exhaustive search for the optimal hyperparameter which minimizes the expectation test of the leave-one-out (loo) error rate. We take existing VC theoretical bounds on the generalization error for SVMs instead of performing loo cross-validation.

We briefly review the VC theoretical bounds here for the sake of completeness. The principle topic of statistical learning theory is the approach of learning by example, which can be described as follows. Assume that there exist l random, independent, identically distributed examples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_l, y_l) \in (R^n, R)$, drawn from a uniform probability distribution $P(\mathbf{x}, y)$ ($P(\mathbf{x}, y) = P(\mathbf{x})P(y | \mathbf{x})$). Given a set of functions $f(\mathbf{x}, \alpha)$, $\alpha \in \Lambda$ (where Λ is a parameter set), then the goal of learning by example is to select a function $f(\mathbf{x}, \alpha_0)$ that can express the relationship between $\mathbf{x}$ and y in the best possible way. In general, in order to obtain $f(\mathbf{x}, \alpha_0)$ one has to minimize the expected risk functional

$$R(\alpha) = \int L(y, f(\mathbf{x}, \alpha)) P(\mathbf{x}, y) d\mathbf{x} dy, \quad (2)$$

where $L(y, f(\mathbf{x}, \alpha))$ measures the classification loss between a response y and a given input $\mathbf{x}$, and the response $f(\mathbf{x}, \alpha)$ is provided by the learning machine.

Learning problems such as pattern recognition, regression estimation, and density estimation are all equivalent to this learning model, for different choices of the loss function [26]. In this paper, we only deal with the classification problem. Consider the following loss function:

$$L(y, f(\mathbf{x}, \alpha)) = \begin{cases} 0 & \text{if } y = f(\mathbf{x}, \alpha) \\ 1 & \text{if } y \neq f(\mathbf{x}, \alpha) \end{cases} \quad (3)$$

Because the probability distribution $P(\mathbf{x}, y)$ in Eq. (2) is unknown, the expected risk functional is replaced by the empirical risk functional

$$R_{emp}(\alpha) = \frac{1}{l} \sum_{i=1}^l L(y_i, f(\mathbf{x}_i, \alpha)). \quad (4)$$

In order to measure the empirical risk $R_{emp}(\alpha)$ and thereby approximate the expected risk, Vapnik presents the following bound theorem [26]. With probability at least $1 - \eta$ ($0 \leq \eta \leq 1$), the inequality

$$R(\alpha) \leq R_{emp}(\alpha) + \frac{\varepsilon}{2} \left[1 + \sqrt{1 + \frac{4R_{emp}(\alpha)}{\varepsilon}} \right] \quad (5)$$

is simultaneously true for all functions in the set. In Eq. (5), $\varepsilon = (4/l)(h(\ln(2l/h)) + 1 - \ln \eta)$ and h is the VC dimension of the set of functions $f(\mathbf{x}, \alpha)$, $\alpha \in \Lambda$.

From Eq. (5), we can see that minimizing the expected risk $R(\alpha)$ is equivalent to minimizing both terms on the right-hand side of Eq. (5) at the same time. The first term, $R_{emp}(\alpha)$, is minimized by the learning process. The second term varies with the VC dimension h and the number of examples l . The smaller the VC dimension and the larger the number of examples, the smaller the second term will be. The number of examples is, of course, finite.

In order to solve the problem we therefore need to construct a set of functions. The classical Lagrangian duality can be used to transform the primal problem into an equivalent dual problem such as Eq. (1). The decision function then has the following form:

$$f(\mathbf{x}) = \sum_{i=1}^l \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}_j) + b, \text{ and } y = \text{sign}(f(\mathbf{x})). \quad (6)$$

In this equation, $\alpha(\alpha_1, \alpha_2, \dots, \alpha_l)$ is the Lagrange multiplier. Each sample has its corresponding α_i , $i=1, \dots, l$, but only a small fraction of the α_i coefficients are nonzero. Corresponding pairs of α_i entries are known as *support vectors*, and they fully define the decision function. All other training samples having $\alpha_i = 0$ can be removed from the training set without affecting the optimal hyperplane.

4 Experimental Results

Two DGA datasets were compiled to study the feasibility of this approach. The larger data set is comprised of 635 historical dissolved gas records, and includes a variety of power transformers of the Taipower company. A second sample of 134 gas records from the IEC TC10 Database is classified into the 6 IEC 60599 category descriptions based on actual conditions [27].

We are concerned not only with the three basic gas ratios of DGA data, $\text{C}_2\text{H}_2/\text{C}_2\text{H}_4$, CH_4/H_2 , and $\text{C}_2\text{H}_4/\text{C}_2\text{H}_6$, but also with their mean (μ), root mean square (rms), variance (σ^2), skewness (normalized 3rd central moment γ_3), kurtosis (normalized fourth central moment γ_4), and normalized fifth to tenth central moments (γ_5 – γ_{10}) as follows:

$$\gamma_n = \frac{E\{[y_i - \mu]^n\}}{\sigma^n}, \quad n = 3, 10 \quad (7)$$

where E represents the expected value of the function [22]. The total number of derived features is therefore 36. We randomly extract instances from these datasets to create the training, validation, and test sets for the corresponding phases [27].

4.1 Performance Comparison for the IEC TC 10 Dataset

In this section classification results are presented for straightforward ANNs and SVMs, both with and without feature and parameter selection. For each ANN the number of neurons in the hidden layer was fixed at 36; for each SVM the width (σ) was set to 0.5 and the tradeoff regularization parameter (C) was set to 100. These values were chosen on the basis of initial training trials.

Table 1 shows classification results for the ANN and SVM using the IEC TC 10 dataset, without automatic CSA selection of features and parameters. In this case, all the features from the DGA signals were used. The success rate of the straightforward ANN was 83.6%, and that of the SVM was 90.4%. In this experiment, the training time was much higher for the ANN than for the SVM. Table 2 displays results for the CSA optimization case, where input features were automatically selected from the set of 36 features described above. The remaining features are 12 for ANN and 4 for SVM respectively. The test performance improved substantially when features and

parameters were selected through CSA optimization, for both the ANN and the SVM. The success rate was 96.3% for the ANN, and 99.4% for the SVM.

The computational times (as measured on a PC with a 1.7GHz Pentium IV processor and 512MB RAM) for training these four classifiers are also shown, though a direct comparison is difficult due to differences in each case's code efficiency. These values are mentioned for the purpose of rough comparison only. It should be mentioned, however, that the difference in computational time for training should not be very important, as it can be done off-line.

Table 1. Performance without CSA optimization for the IEC TC 10 dataset

Classifier	Straightforward			
	No. of inputs	Training success(%)	Test success(%)	Training time(s)
ANN	36	78.7	83.6	16.385
SVM	36	81.8	90.4	4.937

Table 2. Performance with CSA optimization for the IEC TC 10 dataset

Classifier	With CSA			
	No. of inputs	Training success(%)	Test success(%)	Training time(s)
ANN	12	89.9	96.3	4.212
SVM	4	97.9	99.4	3.435

4.2 Performance Comparison for the Taipower Company Dataset

This data set is comprised of 635 historical dissolved gas records of power transformers located at transmission network from Taipower Company. Table 3 shows classification test results for the basic ANN and SVM methods using this dataset. The success rates are 82.9% for the ANN, and 89.7% for the SVM. Table 4 shows the results after CSA optimization is performed, demonstrating that the proposed method is capable of improving classification systems by removing redundant and potentially confusing input features. The input features were automatically selected from the range of 36 DGA features described above by the proposed method. The remaining features are 4 for both ANN and SVM respectively. In the ANN case, the number of neurons in the hidden layer was also selected in the range 10–36; for SVMs, the RBF kernel width was also selected in the range 0.1–2.0. The test performance improved substantially with feature and parameter selection for both ANN and SVM: the success rate was 99.7% for the ANN and 100% for the SVM. Note that in all cases the data sets used for testing were not used in constraining the model.

Finally, we mention that values of the kernel parameters σ , ε , and C in the SVM model are also obtained by the above experiments. The most suitable kernel parameters in the CSA-based SVM models are those with the highest level of affinity. Their corresponding values for the different DGA datasets are given in Table 5.

Table 3. Performance without CSA optimization for the Taipower Company dataset

Classifier	Straightforward			
	No. of inputs	Training success(%)	Test success(%)	Training time(s)
ANN	36	77.6	82.9	20.221
SVM	36	79.0	89.7	4.104

Table 4. Performance with CSA optimization for the Taipower Company dataset

Classifier	With CSA			
	No. of inputs	Training success(%)	Test success(%)	Training time(s)
ANN	4	100	99.7	5.335
SVM	4	100	100	2.076

Table 5. Kernel parameters of the CSA-based SVM model

Dataset	SVM kernel parameters		
	σ	C	ε
IEC TC 10	0.3	100	400
Taipower	0.7	70	90

5 Conclusions

In this paper, the selection of appropriate input features and classification parameters has been optimized using a CSA-based approach. A procedure is presented for the detection of power transformer faults for two classifiers, ANNs and SVMs, using CSA-based feature and parameter selection from inherently imprecise DGA signal data. An appropriate VC bound is selected as the objective function for SVM automatic parameter selection in experimental tests. Test results from practical data demonstrate the effectiveness and high efficiency of the proposed approach. The classification accuracy of SVMs was better than that of ANNs when applied without the CSA. With CSA-based selection, the performances of the two classifiers were comparable at nearly 100%. This result shows the potential of CSAs for off-line feature and parameter selection, and opens up the possibility of pre-optimized features and classification parameters in real-time implementations. These developments could lead to the creation of an effective automated DGA condition monitoring and diagnostic system. We conclude that our method is not only able to figure out optimal parameters, but also minimize the number of features that an SVM classifier should process and maximize diagnosis success rates for DGA data of power transformers.

References

1. IEC Publication 599, "Interpretation of The Analysis of Gases in Transformers And Other Oil-Filled Electrical Equipment In Service," First Edition, 1978.
2. "IEEE Guide for The Interpretation of Gases Generated In Oil-Immersed Transformers," IEEE Std.C57.104-1991.
3. R.R. Rogers, "IEEE And IEC Codes to Interpret Faults in Transformers Using Gas in Oil Analysis," IEEE Trans Electron. Insulat. 13 (5), pp.349–354, 1978.
4. M.Dong, "Fault Diagnosis Model Power Transformer Based on Statistical Learning Theory and Dissolved Gas Analysis," IEEE 2004 International Symposium on Electrical Insulation, p85-88.
5. N.C.Wang, "Development of Monitoring and On-line Diagnosing System for Large Power Transformers," (Power Research Institute, TPC) .
6. P. Purkait, S. Chakravorti, An expert system for fault diagnosis in transformers during impulse tests, in: Power Engineering Society Winter Meeting, vol. 3, 23–27 Jan, 2000, pp. 2181–2186..
7. Q. Su, C. Mi, L.L. Lai, P. Austin, A fuzzy dissolved gas analysis method for the diagnosis of multiple incipient faults in a transformer, IEEE Trans. Power Syst. 15 (2) (2000) 593–598.
8. L.B. Jack, A.K. Nandi, "Fault Detection Using Support Vector Machines and Artificial Neural Networks: Augmented by Genetic Algorithms," Mech. Syst. Signal Process. 16 (2–3), pp.373–390, 2002.
9. C.H. Yann, A new data mining approach to dissolved gas analysis of oil-insulated power apparatus, IEEE Trans. Power Deliv. 18 (4)(2003) 1257–1261.
10. Y.C. Huang, H.T. Yang, C.L. Huang, Developing a new transformer fault diagnosis system through evolutionary fuzzy logic, IEEE Trans. Power Deliv. 12 (2) (1997) 761–767.
11. T.Y. Hong, C.L. Chiung, Adaptive fuzzy diagnosis system for dissolved gas analysis of power transformers, IEEE Trans. Power Deliv. 14 (4) (1999) 1342–1350.
12. M.H. Wang, Extension neural network for power transformer incipient fault diagnosis, in: IEEE Proceedings – Generation, Transmission and Distribution, vol. 150, no. 6, 2003, pp. 679–685.
13. C.H. Yann, Evolving neural nets for fault diagnosis of power transformers, IEEE Trans. Power Deliv. 18 (3) (2003) 843–848.
14. C. Cortes, V. Vapnik, Support-vector Networks, Machine Learn. 20 (3) (1995) 273–295.
15. J.W. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, Face recognition using feature optimization and _-support vector learning Neural Networks for Signal Processing XI, in: Proceedings of the 2001 IEEE Signal, Processing Society Workshop, 10–12 September, 2001, pp.373–382.
16. E.H. Tay Francis, L.J. Cao, Application of support vector machines in financial time series forecasting, Omega. 29 (4) (2001) 309–317.
17. W.C. Chan, C.W. Chan, K.C. Cheung, C.J. Harris, On the modeling of nonlinear dynamic systems using support vector neural networks, Eng. Appl. Artif. Intell. 14 (2001) 105–113.
18. Vapnik V, Golowich S, Smola A. "Support Vector Method for Function Approximation, Regression Estimation, And Signal Processing ." In: Mozer M, Jordan M, Petsche T (eds). Neural Information Processing Systems . MIT Press, 1997, 9.
19. O. Chapelle and V. Vapnik. Model selection for Support Vector Machines. In S. Solla, T. Leen, and K.-R. Muller, editors, *Advances in Neural Information Processing Systems 12*, MA, MIT Press, 2000.

20. O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing Multiple Parameters for Support Vector Machines. *Machine Learning*, 46(1):131 – 159, 2002.
21. C. Cortes, V. Vapnik, “Support Vector Networks,” *Mach Learning* 20 (3) pp.273–295, 1995.
22. H. Frohlich. Feature Selection for Support Vector Machines by Means of Genetic Algorithms. Master’s thesis, University of Marburg, 2002.
23. de Castro, L. N. & Von Zuben, F. J. (1999), “Artificial Immune Systems: Part I – Basic Theory and Applications”, *Technical Report – RT DCA 01/99*, pp. 90.
24. W.W. Yan, H.H. Shao, “Application of Support Vector Machine Nonlinear Classifier to Fault Diagnoses,” *Proceedings of the Fourth World Congress Intelligent Control and Automation*, 10–14 Shanghai, China, pp. 2697–2670, June 2002.
25. J. H. Lee, and C. J. Lin, Automatic model selection for support vector machines, Technical Report, Dept. of Computer Science and Information Engineering, Taipei, Taiwan, November 2000.
26. V.N. Vapnik, “The Nature of Statistical Learning Theory,” Springer-Verlag, New York, 1995.
27. Michel Duval, ”Interpretation of Gas-In-Oil Analysis Using New IEC Publication 60599 and IEC TC 10 Databases,” *IEEE Electrical Insulation Magazine*, March/April 2001 — Vol. 17, No. 2.

An Overview of Alternative Rule Evaluation Criteria and Their Use in Separate-and-Conquer Classifiers

Fernando Berzal, Juan-Carlos Cubero, Nicolás Marín, and José-Luis Polo

Department of Computer Science and Artificial Intelligence,
University of Granada, 18071 Granada, Spain
{fberzal, JC.Cubero, nicm, jlpolo}@decsai.ugr.es

Abstract. Separate-and-conquer classifiers strongly depend on the criteria used to choose which rules will be included in the classification model. When association rules are employed to build such classifiers (as in ART [3]), rule evaluation can be performed attending to different criteria (other than the traditional confidence measure used in association rule mining). In this paper, we analyze the desirable properties of such alternative criteria and their effect in building rule-based classifiers using a separate-and-conquer strategy.

1 Introduction

The aim of any classification algorithm is to build a classification model given some examples of the classes we are trying to model. The model we obtain can then be used to classify new examples or simply to achieve a better understanding of the available data. Rule-based classifiers, in particular, can be built using a “Separate and Conquer” strategy (as in decision lists) or a “Divide and Conquer” approach (as in decision trees). In the former case, the criteria used to evaluate the rules that will be included in the classification model are of the utmost importance.

In this paper, we analyze alternative rule evaluation criteria and study their actual impact in “separate and conquer” classification models built with ART [3], which is a generalized “separate and conquer” algorithm that builds decision lists that can also be viewed as degenerate, polythetic decision trees.

2 Alternative Rule Evaluation Criteria

Even though accurate classification models can be built using standard association rules [3] [1] [17] [11] [14] [18] [15] [7] [8] [12] [13] [10] [19], it is clear that confidence is not the best measure for a classification rule. Alternative rule evaluation measures might provide better insight into the capability of a given association rule to classify data. Before delving into the details of existing measures, we should review what we consider a good classification rule. An association rule $A \Rightarrow C$ will be useful in classification problems when the logical implication $A \rightarrow C$ is valid:

- C happens more often when A holds.
- $\neg C$ should be less frequent when A holds.

We might also be interested in verifying the validity of $\neg C \rightarrow \neg A$, which is mathematically equivalent to $A \rightarrow C$. Hence, a potentially useful rule should also verify the following properties:

- $\neg A$ happens more often when $\neg C$.
- A should occur less frequently when C does not hold.

Although the latter two properties might be interesting from a logical point of view, they do not directly affect classification accuracy, where we are interested only in determining the class C given A . Even then, those properties might be desirable to improve the understandability of the classification model obtained.

The following paragraphs discuss some rule evaluation measures and their properties in the context of classification problems.

Confidence

Even when the rule support might be of interest during the association discovery process from a purely tactical point of view, the rule confidence is what finally determines the rule validity. Its meaning is easy to grasp, and it can be defined as follows: $\text{conf}(A \Rightarrow C) = P(C|A) = P(A \cap C)/P(A)$. Confidence is used to measure the association between the antecedent A and the consequent C : the higher the confidence of $A \Rightarrow C$, the lower the confidence of $A \Rightarrow \neg C$, since $P(C|A) + P(\neg C|A) = 1$. However, the rule confidence cannot be used to establish a causality relationship between antecedent and consequent. It cannot be used to perform inferences since it does not take into account the validity of $\neg C \Rightarrow \neg A$. A variation of the confidence does, hence its name: Causal confidence [9].

Causal Confidence

The causal confidence measure considers the confidence of the rule $A \rightarrow C$ and the confidence of its counterpart $\neg C \rightarrow \neg A$. Thus, the definition is given by $\text{conf}_{\text{causal}}(A \Rightarrow C) = \frac{1}{2}(P(C|A) + P(\neg A|\neg C))$. The average is used just to normalize the measure so that its values are always in the interval $[0, 1]$. Unfortunately, this measure could have a high value even when the implication $A \rightarrow C$ does not have a representative support but its counterpart $\neg C \rightarrow \neg A$ does. Therefore, causal confidence is not adequate for classification problems.

Causal Support

The idea of causal confidence can also be applied to the support measure in order to obtain the causal support: $\text{support}_{\text{causal}}(A \Rightarrow C) = P(A \cap C) + P(\neg A \cap \neg C)$. As before, even when $P(A \cap C)$ is really low, $P(\neg A \cap \neg C)$ can be high, causing causal support to be high (even when the rule might not be too useful in a classification problem). Consequently, causal support is not a good choice to evaluate rules in classification problems.

Confirmation

Another measure, called confirmation, takes into account when the antecedent holds but not its consequent, what might be used to highlight a rule depending

on the commonness of its antecedent. $K(A \Rightarrow C) = P(A \cap C) - P(A \cap \neg C)$. Since $P(A \cap \neg C) = P(A) - P(A \cap C)$, confirmation is reduced to $P(A \cap C) - P(A \cap \neg C) = 2P(A \cap C) - P(A)$. Since $P(A \cap C)$ is the rule support and $P(A)$ is the support of the antecedent, confirmation is no more than $K(A \Rightarrow C) = \text{support}(A \Rightarrow C) - \text{support}(A)$. Hence

$$K(A \Rightarrow C) \leq \text{support}(A \Rightarrow C) \quad (1)$$

The second term just penalizes the support of the rule according to the support of the antecedent. In other words, given two equally common rules, confirmation would select as most interesting the rule whose antecedent is less common. That might be useful in the knowledge discovery process, although it does not help in classification problems because it does not take into account relationship between the rule antecedent and the rule consequent. This measure has been used to obtained three new criteria to evaluate rules:

- **Causal confirmation**, takes into account $\neg C \rightarrow \neg A$ and can be reduced to $K_{\text{causal}}(A \Rightarrow C) = \text{support}_{\text{causal}}(A \Rightarrow C) - \text{support}(A) + \text{support}(C)$. As the standard confirmation, this measures varies depending upon the support of A . On the other hand, the more common the class C is, the higher causal confirmation the rule will have, what certainly makes classification difficult (especially when the class distribution is skewed). Apart from this fact, causal support is not suitable for classification problems.
- **Confirmed confidence** is obtain from the standard confidence when we try to measure the quality of the rule $A \Rightarrow \neg C$ (that is, when the antecedent holds but not the consequent): $\text{conf}_{\text{confirmed}}(A \Rightarrow C) = P(C|A) - P(\neg C|A)$. However, since $P(\neg C|A) = 1 - P(C|A)$, confirmed confidence can be reduced to $\text{conf}_{\text{confirmed}}(A \Rightarrow C) = 2 \cdot \text{conf}(A \Rightarrow C) - 1$. Therefore, if we are using this measure just to rank the candidate rules which might be part of a classification model, the rule confirmed confidence is completely equivalent to the rule confidence (only that the confirmed confidence is defined over the $[-1, 1]$ interval).
- Even a **causal confirmed confidence** can be concocted from the two previous measures, but no interesting results can be expected when dealing with classification problems.

Conviction

Conviction [4] was introduced as an alternative to confidence to mine association rules in relational databases (implication rules using their authors' nomenclature). $\text{conviction}(A \Rightarrow C) = \frac{P(A)P(\neg C)}{P(A \cap \neg C)}$. Similar in some sense to confirmation, conviction focuses on $A \Rightarrow \neg C$:

$$\text{conviction}(A \Rightarrow C) = \frac{\text{support}(\neg C)}{\text{conf}(A \Rightarrow \neg C)} \quad (2)$$

The lower $P(A \cap \neg C)$, the higher the rule conviction, what makes conviction ideal for discovering rules for uncommon classes. However, the conviction domain is

not bounded and it is hard to compare conviction values for different rules. This constrains the usability of the conviction measure in classification models such as ART [3], since no heuristics to automatically settle a conviction threshold can be devised.

Later in this paper, another bounded measure will be analyzed which is completely equivalent to conviction in the situations of interest for classification problems: the well-known Shortliffe and Buchanan's certainty factors. When there are at least two classes (i.e. $\text{support}(C) < 1$) and the rule improves classifier accuracy (i.e. $CF(A \Rightarrow C) > 0$), certainty factors can be defined as

$$CF(A \Rightarrow C) = 1 - \frac{1}{\text{conviction}(A \Rightarrow C)}$$

This allows us to substitute conviction for another measure whose domain is bounded. Further properties of the certainty factor will be discussed in 2.

Interest

A rule interest measure was defined in [16] as $\text{interest}(A \Rightarrow C) = \frac{P(A \cap C)}{P(A)P(C)} = \frac{P(C|A)}{P(C)}$. The more common A and C , the less interest the rule will have, which is certainly useful to guide the knowledge discovery process. Among its properties, its symmetry stands out: the interest of $A \Rightarrow C$ equals to the interest of $C \Rightarrow A$. As happened with conviction, its domain is not bounded, what might make its interpretation harder in a classification model. In some sense, it can be considered complementary to conviction if we take into account the following equality and compare it with equation 2, although interest focuses on the association $A \Rightarrow C$ while conviction focuses on $A \Rightarrow \neg C$:

$$\text{interest}(A \Rightarrow C) = \frac{\text{confidence}(A \Rightarrow C)}{\text{support}(C)} \quad (3)$$

Dependency

The following measure is the discrete equivalent of correlation in continuous domains. $\text{dependency}(A \Rightarrow C) = |P(C|A) - P(C)|$. In classification problems, it is not suitable since its value is high for common classes even when the corresponding rule confidence is minimum: $\text{dependency}(A \Rightarrow C) = |\text{conf}(A \Rightarrow C) - \text{support}(C)|$. A causal variation [9] of this measure can also be defined as follows, although it is not useful in classification problems either. Bhandari's attribute focusing measure is also derived from the dependency measure above, as the following expression shows $\text{Bhandari}(A \Rightarrow C) = \text{support}(A) \cdot \text{dependency}(A \Rightarrow C)$. Therefore, it is of no use in classification problems.

Hellinger's Divergence

Hellinger's divergence was devised to measure the amount of information a rule provides [5] and it can be viewed as a distance measure between a priori and a posteriori class distributions:

$$H(A \Rightarrow C) = \sqrt{P(A)}[(\sqrt{P(A \cap C)} - \sqrt{P(C)})^2 - (\sqrt{1 - P(A \cap C)} - \sqrt{1 - P(C)})^2]$$

This measure has been used in classifiers before and will be evaluated in Section 3.

Certainty Factors

Certainty factors were introduced by Shortliffe and Buchanan to represent uncertainty in the MYCIN expert system. Its use in association rule mining has been proposed in [2]. The certainty factor of a rule $A \Rightarrow C$ is defined as

$$CF(A \Rightarrow C) = \frac{conf(A \Rightarrow C) - support(C)}{1 - support(C)}$$

when $conf(A \Rightarrow C) > support(C)$,

$$CF(A \Rightarrow C) = \frac{conf(A \Rightarrow C) - support(C)}{support(C)}$$

when $conf(A \Rightarrow C) < support(C)$, and

$$CF(A \Rightarrow C) = 0$$

when $conf(A \Rightarrow C) = support(C)$. This rule evaluation measure can be viewed as the variation degree of the probability of C when A holds. The larger a positive certainty factor, the smaller the decrease of the probability of C not being when A holds. In extreme situations, the rule confidence determines its certainty factor:

$$conf(A \Rightarrow C) = 1 \Rightarrow CF(A \Rightarrow C) = 1$$

$$conf(A \Rightarrow C) = 0 \Rightarrow CF(A \Rightarrow C) = -1$$

Certainty factor take into account the probability of C apart from the rule confidence. They also verify an interesting property when they are positive (which is when the rules are useful for classification):

$$CF(A \Rightarrow C) = CF(\neg C \Rightarrow \neg A) \quad (4)$$

In classification problems, we could face different situations where certainty factors behavior are at their best:

1. If our problem includes a skewed class distribution, and two candidate rules hold the same confidence value but correspond to classes of different frequency, the rule corresponding to the less common class has a higher certainty factor:

$$\begin{aligned} conf(A \Rightarrow C) = conf(B \Rightarrow D) \leq 1, \quad support(C) > support(D) \rightarrow \\ \rightarrow CF(A \Rightarrow C) \leq CF(B \Rightarrow D) \end{aligned}$$

2. Under some circumstances, comparing certainty factors is equivalent to comparing confidence values. $CF(A \Rightarrow C_1) > CF(B \Rightarrow C_2)$ can be reduced to $conf(A \Rightarrow C_1) > conf(B \Rightarrow C_2)$ when

- Both rules refer to the same class c_k .
 - Both rules correspond to classes with the same probability: $support(c_1) = support(c_2)$.
3. Finally, there exist situations where higher certainty factors do not correspond to a higher confidence values. Given two rules so that $CF(A \Rightarrow C) > CF(B \Rightarrow D)$:
- When C is more common than D:

$$conf(A \Rightarrow C) > K \cdot conf(B \Rightarrow D) \quad , \quad K = \frac{support(\neg C)}{support(\neg D)} < 1$$

- When C is less common than D:

$$conf(A \Rightarrow C) > conf(B \Rightarrow D) - \Delta \quad , \quad \Delta = \frac{support(D) - support(C)}{support(\neg C)} > 0$$

In summary, even though certainty factors are intimately linked to confidence values, a higher certainty factor does not imply a higher confidence value:

$$CF(A \Rightarrow C) > CF(B \Rightarrow D) \Rightarrow conf(A \Rightarrow C) > conf(B \Rightarrow D)$$

The relative frequency of each class determines the higher certainty factors. Let us suppose that we have two association rules: $A \Rightarrow c_1$ with 2% support and 50% confidence, and $B \Rightarrow c_2$ with 50% support and 95% confidence. The certainty factors of such rules would be 3/8 and 3/4 if c_1 has a 20% support and c_2 has a 80% support. However, if the class distribution varies, being now 10% of c_1 and 90% of c_2 , when certainty factors would be 4/9 and 1/2, keeping their relative order. However, if the class distribution is inverted and now c_1 has a 94% support while c_2 only has a 6% support, when certainty factors would become 46/94 and 1/6, being the second lower than the first!

Situations such as the one described in the paragraph above should be used as a warning sign when comparing certainty factors to choose the rules to include in a classification model. Sometimes they might be useful, as when they prefer rules corresponding to uncommon classes, although you should also expect shocking difference between classification models when class distributions change.

2.1 A Usefulness Constraint

Certainty factor properties suggest an additional pruning step when considering association rules in order to build classification models. When you build such models, only association rules with a positive certainty factor are really useful, since they increase our knowledge and improve classifier accuracy. By definition, a positive certainty factor is obtained for a rule $A \Rightarrow C$ when $conf(A \Rightarrow C) > support(C)$. This constraint indicates that the use of the rule $A \Rightarrow C$ improves the classification model which would result from using a default class (at least with respect to C in the training set). That idea can be used to prune association rules which do not verify the above requirement, which can be expressed as $P(A \cap C) > P(A) \cap P(C)$. That rule pruning can be viewed as a ‘usefulness criterion’ which reduces the number of candidate association rules which can become part of the classification model

3 Experimental Results Using ART

Some experiments have been performed to check the influence of the rule evaluation measure on the process of building a classification model. We have tested different criteria using the ART [3] classification model as a test bed, since ART does not require any specific measure to evaluate the rules the ART classifier is built from. Our experiments try to estimate the suitability of the measures proposed in the previous section, using 10-folded cross validation and the same datasets which were used in [3]. In all our experiments with ART, we used a 5% minimum relative support threshold and the automatic threshold selection heuristics described in [3]. Table 1 summarizes our results.

The following observations can be made from the results we have obtained:

- The usefulness criterion proposed in 2.1 consistently improves ART classification accuracy. Moreover, it improves accuracy without modifying the evaluation measure used during the association rule mining process (that is, the rule confidence). However, this increased accuracy comes at a cost: the increased complexity of the resulting classifier. The resulting ART tree has more leaves and, therefore, training time is somewhat higher since the training dataset must be scanned more times to build the classifier. This result is just an incarnation of the typical trade-off between classifier accuracy and classifier complexity.
- Certainty factors do not improve ART overall performance, probably due to some of their counterintuitive properties (see section 2).
- As we could expect from the properties analyzed in the previous section, the use of conviction achieves results which are similar to the results obtained by using certainty factors. From a classification point of view, conviction and certainty factors are equivalent when it is interesting to include a given association rule in the classification model, as was mentioned in section 2.
- The use of the interest measure (section 2) leads to the results we could expect in ART: since this measure is not bounded, the automatic threshold selection criterion in ART does not work properly. Perhaps, the additive tolerance margin might be replaced by a multiplicative tolerance factor. Even then, the definition of the interest measure makes it difficult to establish an initial desirable interest value. Such value might depend on the particular problem and, therefore, the use of the interest measure is not practical in ART. Bounded measures, such as confidence or certainty factors, will be preferred.
- The same rationale applies to Hellinger's divergence (section 2). Even when its range is bounded, the interpretation and comparison of Hellinger's divergence values make this measure impractical in classification models such as ART. In fact, no acceptable classification models were built by ART because it is hard to establish an initial desirable value for Hellinger's divergent (a prerequisite to make use of ART automatic parameter setting).

Table 1. Experiment summary using different rule evaluation criteria

	Confidence Usefulness		CF	Conviction	Interest	Hellinger
Accuracy (10-CV)	79.22%	83.10%	77.67%	77.79%	53.71%	48.05%
Training time	18.5s	22.4s	12.7s	10.7s	2.9s	1.8s
Tree topology						
- Leaves	36.9	50.0	28.5	30.0	4.2	1
- Internal nodes	18.3	25.2	16.2	17.3	2.6	0
- Average depth	7.41	8.71	7.39	7.33	2.12	1.00
I/O operations						
- Records	36400	50100	33100	26700	20300	7000
- Scans	63	89	55	59	11	3

Table 2. Datasets used in our experiments (from the UCI Machine Learning Repository)

<i>Dataset</i>	<i>Records</i>	<i>Attr</i>	<i>Classes</i>
AUDIOLOGY	226	70	24
CAR	1728	7	4
CHESS	3196	36	2
HAYES-ROTH	160	5	3
LENSES	24	6	3
LUNG CANCER	32	57	3
MUSHROOM	8124	23	2
NURSERY	12960	9	5
SOYBEAN	683	36	19
SPLICE	3175	61	3
TICTACTOE	958	10	2
TITANIC	2201	4	2
VOTE	435	17	2

4 Conclusions

In summary, from all the measures we discussed above, only certainty factors and the so-called usefulness criterion are good alternatives to confidence when building ART classification models. Conviction also achieves good results, although certainty factors are preferred since they are equivalent to conviction in the cases the classification process is more interested in (and certainty factors are bounded).

Despite the experimental results, it should be noted that all rule evaluation criteria have their home grounds and their use might be suitable depending on what the user intends to obtain. The availability of a wide variety of rule evaluation measures is a good signal, since it provides us a toolkit to draw on.

Moreover, the use of a measure or another does not affect the computational cost of the rule discovery process. When building classification models such as ART, that cost is proportional to the classification model complexity. Therefore, the study of alternative rule evaluation measures keeps its interest. Such measures are just criteria at our disposal which can be used to guide the knowledge discovery process according to the particular goals and needs of a given problem.

References

1. K. Ali, S. Manganaris, and R. Srikant. Partial classification using association rules. In *Proceedings of the 3rd International Conference on Knowledge Discovery in Databases and Data Mining, Newport Beach, California, USA*, pages 115–118, August 14–17 1997.
2. Fernando Berzal, Ignacio Blanco, Daniel Sanchez, and M.A. Vila. Measuring the accuracy and interest of association rules: A new framework. *Intelligent Data Analysis*, 6(3):221–235, 2002.
3. Fernando Berzal, Juan Carlos Cubero, Daniel Sánchez, and J.M. Serrano. Art: A hybrid classification model. *Machine Learning*, 54(1):67–92, January 2004.
4. S. Brin, R. Motwani, J.D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the ACM SIGMOD international conference on Management of Data, Tucson, Arizona, USA*, pages 255–264, May 11–15 1997.
5. Lee C.-H. and Shim D.-G. A multistrategy approach to classification learning in databases. *Data & Knowledge Engineering*, 31:67–93, 1999.
6. W. Cohen. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning, Tahoe City, CA USA*, pages 115–123, July 1995.
7. G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge Discovery and Data Mining, San Diego, CA USA*, pages 43–52, August 15–18 1999.
8. G. Dong, X. Zhang, L. Wong, and J. Li. Caep: Classification by aggregating emerging patterns. In *Proceedings of the Second International Conference on Discovery Science, Tokyo, Japan*, pages 30–42, 1999.
9. Y. Kodratoff. Comparing machine learning and knowledge discovery in databases. *Lecture Notes in Artificial Intelligence, LNAI*, 2049:1–21, 2001.
10. Wenmin Li, Jiawei Han, and Jian Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM'01), San Jose, California, USA*, pages 208–217, November 29 - December 02 2001.
11. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, USA*, pages 80–86, August 27–31 1998.
12. B. Liu, M. Hu, and W. Hsu. Intuitive representation of decision trees using general rules and exceptions. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000), Austin, Texas*, pages 30–42, July 30 - August 3 2000.
13. B. Liu, M. Hu, and W. Hsu. Multi-level organization and summarization of the discovered rule. In *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA USA*, pages 208–217, August 20 - 23 2000.

14. B. Liu, Y. Ma, and C.K. Wong. Improving an association rule based classifier. In *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000)*, Lyon, France, pages 80–86, September 13–16 2000.
15. D. Meretakakis and B. Wuthrich. Extending naive bayes classifiers using long itemsets. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge Discovery and Data Mining, San Diego, CA USA*, pages 165–174, August 15–18 1999.
16. C. Silverstein, S. Brin, and R. Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 22(2):39–68, 1998.
17. K. Wang, S. Zhou, and Y. He. Growing decision trees on support-less association rules. In *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA USA*, pages 265–269, August 20–23 2000.
18. K. Wang, S. Zhou, and S.C. Liew. Building hierarchical classifiers using class proximity. In *Proceedings of 25th International Conference on Very Large Data Bases (VLDB'99)*, Edinburgh, Scotland, UK, pages 363–374, September 7–10 1999.
19. Xiaoxin Yin and Jiawei Han. Cpar: Classification based on predictive association rules. In *Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, CA, USA*, pages 208–217, May 1-3 2003.

Learning Students' Learning Patterns with Support Vector Machines

Chao-Lin Liu

Department of Computer Science, National Chengchi University, Taiwan
chaolin@nccu.edu.tw

Abstract. Using Bayesian networks as the representation language for student modeling has become a common practice. Many computer-assisted learning systems rely exclusively on human experts to provide information for constructing the network structures, however. We explore the possibility of applying mutual information-based heuristics and support vector machines to learn how students learn composite concepts, based on students' item responses to test items. The problem is challenging because it is well known that students' performances in taking tests do not reflect their competences faithfully. Experimental results indicate that the difficulty of identifying the true learning patterns varies with the degree of uncertainty in the relationship between students' performances in tests and their abilities in concepts. When the degree of uncertainty is moderate, it is possible to infer the unobservable learning patterns from students' external performances with computational techniques.

1 Introduction

Providing satisfactory interaction between human and machines requires good computational models of human behaviors. Take computer-adaptive testing (CAT) for example. Researchers build models based on the Item-Response Theory (IRT) [1,2] and Concept Maps [3] for predicting students' performances and selecting appropriate test items for assessment. With good student models, a computational system can evaluate students' competence with less test items than traditional paper-and-pencil tests will need, and can achieve better accuracy in its evaluation. In addition, test takers can access a CAT system almost any time at any location, and can obtain their scores on the spot. Hence, CAT has been adopted in many official evaluation activities, including TOFEL and GRE, although there are sporadic criticisms [4].

Typically, domain experts provide information about student models, which are then implemented with computational techniques. CAT systems that adopt IRT assume that a student's responses to test items are mutually independent given the student's competence, so IRT-based systems generally take the so-called naïve Bayes models [5, 6]. Based on this assumption, the problem of building student models boils down to learning the model parameters from observed data [7, 8]. Similarly, Liu et al. assume the availability of concept maps of students and teachers, and design algorithms for comparing the concept maps for assessment [3].

Although human experts can choose good models from candidate models, they may not agree on their choices. For instance, Millán and Pérez-de-la-Cruz discussed a

hierarchical structure of Bayesian networks [9] that included nodes for *subjects*, *topics*, *concepts*, and *questions* [10]. Vomlel employed nodes for *skills* and *misconceptions*, and used relevant nodes as direct parents of nodes for *tasks* [11].

In this paper, we explore computational techniques for comparing the candidate models for students. Although we do not expect computational techniques will give better model structures than human experts will do in the short term, we hope that computational techniques can assist human experts to identify more precise models. More specifically, we would like to guess how students learn composite concepts. A *composite* concept results from students' integration of multiple *basic* concepts. Let $dABC$ denote the composite concept that involves three basic concepts cA , cB , and cC . How do we know how students learn the composite concept? Do they learn $dABC$ by directly integrating the three basic concepts, or do they first integrate cA and cB into an intermediate product, say dAB , and then integrate dAB with cC ?

We compare candidate models that are represented with Bayesian networks, based on students' responses to test items. Students' responses to test items reflect their competences in the tested concepts in an indirect and uncertain manner. The relationship is uncertain because students may make inadvertent errors and luckily hit the correct answers. We refer to these situations as *slip* and *guess*, respectively, henceforth. *Slip* and *guess* are frequently cited in the literature, and many researchers adopted Bayesian networks to capture the uncertainty in their CAT system, e.g., [6, 8, 10-12].

As a result, our target problem is an instance of learning Bayesian networks. This is not a new research problem, and a good tutorial is already available [13]. However, learning Bayesian networks for student modeling is relatively rare, based on our knowledge, particularly when we would try to induce a network directly from students' item responses. Vomlel created network structures from students' data and applied principles provided by experts to refine the structures [11]. Besides those difficulties for learning structures from data, learning a Bayesian network from students' data is more difficult because most of the variables of interests are not directly observable. Hence, the problem involves not just missing values and not just one or two hidden variables.

In our experiments, we have 15 basic and composite concepts. We cannot observe whether students are competent in these concepts directly, though we assume that we can collect students' responses to test items that are related to these concepts. The problem of determining how students learn composite concepts is equivalent to learning the structure of the hidden variables given students' item responses.

We propose mutual information (MI) [14] based heuristics, and apply the heuristics for predicting the hidden structures in two ways: a direct application and training support vector machines (SVMs) [15] for the prediction task. Experimental results indicate that it is possible to figure out the hidden structures under moderate uncertainty between students' item responses and students' competence.

We provide more background information in Section 2, introduce the MI-based heuristics in Section 3, present the SVM-based method in Section 4, and wrap up this paper with a discussion in Section 5.

Table 1. One of the Q-matrices used in the experiments

group	cA	cB	cC	cD	dAB	dAC	dAD	DBC	DBD	dCD	dABC	dABD	dACD	dBCD	dABCD
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	0	0	0	0	1	1	1	1	0	1
3	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1
4	1	1	1	1	1	0	0	0	0	1	1	1	0	0	1
5	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1
6	1	1	1	1	1	0	0	0	0	1	1	0	1	0	1
7	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1
8	1	1	1	1	1	0	0	0	0	1	1	0	0	0	1
9	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1
10	1	1	1	1	1	0	0	0	0	0	1	0	1	1	0
11	1	1	1	1	1	1	0	0	1	1	0	1	0	1	1
12	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1
13	1	1	1	1	0	0	1	1	0	0	0	0	1	1	1
14	1	1	1	1	1	0	0	0	0	1	0	0	1	0	1
15	1	1	1	1	0	0	0	0	1	1	0	0	0	1	1

2 Preliminaries

We provide more formal definitions, explain the source of the simulated students' item responses, and analyze the difficulties of the target task in this section.

The goal of our work is to find the hidden structure of the unobservable nodes that represent students' competence in concepts, based on observed students' item responses. We assume that students learn composite concepts from *parent concepts* that do not have overlapping basic concepts. For the problem of investigating how students learn *dABC* that we mentioned in Section 1, we assume that there are four possible answers: *AB_C*, *AC_B*, *BC_A*, and *A_B_C*, where the underscores separate the parent concepts. Although there is no good reason to exclude a learning pattern like *AB_BC*, including such overlapping parent concepts will dramatically make the problem more complex. We obtained students' item responses from a simulation program that was reported in a previous work [6]. In this paper, we will try to learn how students learn *dABCD*, and there are 14 possible ways to learn this target concept.

2.1 Creating Simulated Students

Although not using real students' data subjects our work to criticisms, we believe that, if we can employ computational models to predict students' behaviors in CAT systems, we should believe that the same computational model is trustworthy for simulating students' behaviors. The use of simulated students is not our invention, previous and well-known work has taken the same approach for studying computational methods, e.g., [10, 12].

Using Liu's simulator that is described in [6], we can control the structure of the Bayesian network and the generation of the conditional probability tables (CPTs). The generation of the CPTs relies on a random number generator that uniformly samples numbers from a given range. In order to specify competence patterns of the student population, we also have to provide a matrix that is similar to the Q-matrix [16], and Table 1 shows the matrix that we used in many of our experiments. The columns are

concepts, and the rows are student groups. When the column is a basic concept, cells are 1 if a typical student in the student group is competent in the concept, and cells are 0 otherwise. When the column is a composite concept, cells are 1 if a typical student in the student

group is able to integrate the parent concepts to form the composite concept if s/he is competent in the parent concepts. Although the cells are either 0 or 1, Liu employed random numbers to give an uncertain relationship between student groups and competence in concepts through a simulation parameter: *groupInfluence*. A student's behavior may deviate from his/her typical group competence pattern with a probability that is uniformly sampled from the range $[0, \text{groupInfluence}]$.

We assumed that every concept had three test items in the experiments, and the network shown in Figure 1 shows a possible network when we consider the problem in which there are only three basic concepts. Note that in this network, we assume that students learn *dABC* by directly integrating the three basic concepts, which is indicated by the direct links from the basic concepts to the node labeled *dABC*. We cannot show the network for the case in which there are four basic concepts in this paper, due to the size of the network. (There will be 15 nodes for concepts, 3×15 nodes for test items, and a lot more links between these 60 nodes.)

The probabilities of *slip* and *guess* are also controlled by a simulation parameter: *fuzziness*. Students of a student group may deviate from the typical behavior with a probability that is uniformly sampled from $[0, \text{fuzziness}]$.

Given the network structure, the Q-matrix, and the simulation parameters, we can create simulated students. In our experiments, we assumed that a student can belong to any of the 16 student groups with equal probabilities. Following Liu's strategy, we used a random number, ρ that was sampled from $[0, 1]$ to determine whether a student would respond to a test item correctly or incorrectly. The conditional probability of correctly responding to a test item given a student belonged to a particular group can be calculated easily with Bayesian networks. Consider the instance for *iA1*, a test item for *cA*. If ρ is smaller than $\Pr(iA1 = \text{correct} \mid \text{group} = g_1)$ when we simulated a student who belonged to the first group, we assumed that this student responded to *iA1* correctly. Since there were 15 concepts, a record for a simulated student would contain the correctness for each of 45 ($=3 \times 15$) test items.

After using the networks to create simulated students, we hid the networks from our programs that took as input the item responses and guessed the structures of the hidden networks.

2.2 Contents of the Q-Matrix and Problem Complexity

The contents of the Q-matrix influence the prior distributions of students' competence patterns and the performance of simulated students [12]. Clearly, there can be many different ways to set the contents of the matrix.

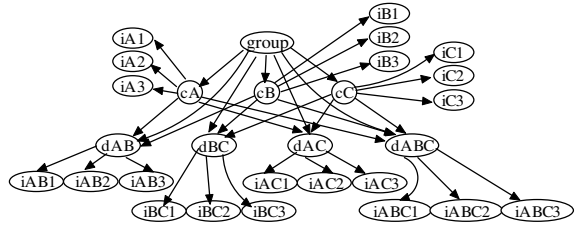


Fig. 1. A Bayesian network for 3 basic concepts

We set the Q-matrix in Table 1, partially based on our experience. Notice that all columns for the basic concepts and the target concept, $dABCD$, are 1. This should be considered a normal choice. If we do want to learn how students learn $dABCD$, we should try to recruit students who appear to be competent in $dABCD$ to participate in our experiments. In addition, there is no good reason to recruit anyone who is not competent in any of the basic concepts in the experiments. We set the values for $dABC$, $dABD$, $dACD$, and $dBCD$ to 16 possible combinations, and this is why we include 16 student groups in Table 1. We randomly choose the values of dXY , where X and Y are symbols for basic concepts, and will report experimental results for other possible settings.

When we consider β basic concepts in the problem, the number of possible ways to learn how students learn a composite concept that is comprised of all these β concepts is related to the Stirling number of the second kind [17]. It is easy to verify that this number grows rapidly with β , and we will have 14 alternatives if we set β to 4.

$$\sum_{i=2}^{\beta} \left(\frac{1}{i!} \sum_{j=0}^{i-1} (-1)^j \binom{i}{j} (i-j)^{\beta} \right) \quad (1)$$

3 MI-Based Heuristics

Consider the situation when we generate students' data from the network shown in Figure 1. If we have the true states of all the concept nodes, it is not difficult to learn the network structure with a variant of the PC algorithm [18] implemented in Hugin [19]. However, we cannot observe the true competence levels of students in reality, and can only indirectly measure the competence through the results of examinations. Moreover, the item responses do not perfectly reflect students' competence, due to many reasons including *guess* and *slip*. Hence, we need to find indirect evidence that may help us to predict the hidden structure.

3.1 Estimating the MI Measures

Recall that we have assumed that every simulated student will respond to three test items for each concept. Out of three test items, a student may correctly respond to 0%, 33%, 67%, and 100% of the test items. Hence, it is possible to estimate the state of the

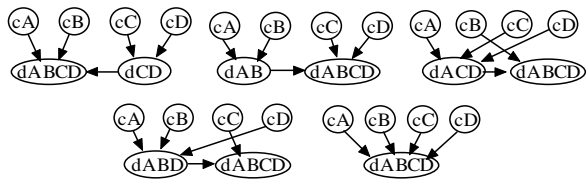


Fig. 2. Candidate (partial) Bayesian networks in our experiments

concept nodes with the percentage of correct responses. We can also use the percentages to estimate the state of a set of variables. For instance, $\Pr(cA=33\%, cB=66\%)$ can be the percentage of students who correctly respond to exactly one item for cA and two items for cB . Given such estimates, we will be able to compute the mutual information between any two sets of variables, and apply the estimated mutual information in guessing the hidden structure.

Intuitively, the variables that are more closely related to each other will exhibit higher mutual information. Partial networks shown in Figure 2 include five possible ways to learn $dABCD$, i.e., A_B_CD , AB_C_D , ACD_B , ABD_C , and $A\_B\_C\_D$. Let $MI(X;Y)$ denote the mutual information between two sets of variables, X and Y . If A_B_CD is the true structure, we expect that it is more likely for the estimated $MI(cA, cB, dCD; dABCD)$ to be larger than the estimated $MI(dAB, cC, cD; dABCD)$ and other estimated MI measures. Hence, we can employ the following heuristics.

Heuristics: The structure that has the largest estimated MI measure is the hidden structure.

Before we estimate the MI measures, we add 0.001 to the number of occurrences of every possible combination of variables. This will avoid the zero probability problems, and is a typical smoothing procedure for estimating probability values [20].

3.2 Experimental Evaluation

Figure 3 shows the flow of how we evaluated the heuristic. In the experiments, we used five different network structures to create simulated students, and their main differences are shown in Figure 2. The parent concepts of other composite concepts that do not appear in the sub-networks in Figure 2 are the basic concepts. For instance, the parent concepts of $dABD$ in the network that used the leftmost sub-network in the top row of Figure 2 are cA , cB , and cD . As we mentioned in Section 2, we mainly used the Q-matrix in Table 1 in our experiments. We set *groupInfluence* and *fuzziness* to different values in $\{0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$, so there were 36 combinations. We did not try values larger than 0.3 because they were beyond consideration normally discussed in the literature. For each of the five network structures and a combination of *groupInfluence* and *fuzziness*, we sampled 600 network instances with the Q-matrix shown in Table 1, and created a different population of 10000 simulated students for each of these instances. The choice of “10000” was arbitrary, and the goal was to make each of the 16 groups include many students.

An *experiment* corresponded to a different combination of *groupInfluence* and *fuzziness*, so there were 36 experiments. We used *accuracy* to measure the quality of our prediction of the hidden structures. It was defined as the percentage of correct prediction of 3000 ($=5 \times 600$) randomly sampled network instances that were used to create the simulated students. According to Equation (1), there were 14 possible answers when β is 4. Hence, to guess the hidden structure of each of these 3000 network instances, we calculated the estimated MI measures for 14 possible answers from the item responses of the 10000 simulated students.

Figure 4 summarizes the experimental results. The vertical axis shows the accuracy, the horizontal axis shows the decimal part of *fuzziness*, and the legends mark the

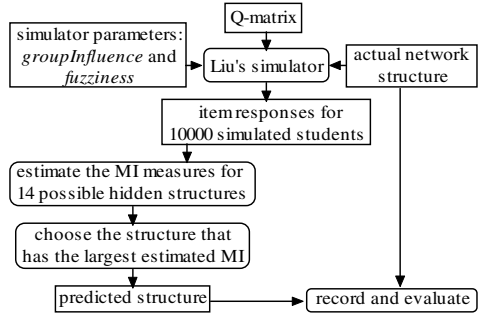


Fig. 3. Flow for evaluating the heuristic

values of *groupInfluence* used in the experiments. Curves in these charts show a general trend that we expected. Increasing the values of *groupInfluence* and *fuzziness* made the relationship between students' item responses and their competence patterns more uncertain and our prediction less accurate. When both *groupInfluence* and *fuzziness* were both close to 0.3, the accuracy was about 0.2.

It is easy to interpret 0.2 as a result of random guesses from five possible answers, but this is not correct. Although we used only five network structures that are shown in Figure 2 to create simulated students, our prediction program did not take this into account, and could consider network structures that were not included in Figure 2. The fact is that our heuristics favored particular structures, which we learned by looking into the internal data collected in experiments. When both *groupInfluence* and *fuzziness* were large, the heuristics tended to favor *A_B_C_D*, which happened to be one of the true answers. As a result, we had the accuracy of 0.2. Had we excluded *A_B_C_D* from the true networks, the accuracy would become smaller than 0.2.

Although we expected that the accuracy should improve as we reduced the values of *groupInfluence* and *fuzziness*, the experimental results did not fully support this intuition. (When we conducted experiments for the cases in which there were only three basic concepts, experimental results did support this intuitive expectation.) When both *groupInfluence* and *fuzziness* were close to 0.05, the heuristics tended to favor *AB_CD* against other competing structures, making the accuracy worse than we expected. More specifically, we created a 14×14 confusion matrix [20] and found that our heuristics chose *AB_CD* relatively frequently when the true structures were *A_B_CD* and *AB_C_D*. The accuracy could hit as low as 0.85 when both *groupInfluence* and *fuzziness* were both 0.05 for other Q-matrices that were different from the Q-matrix shown in Table 1. These Q-matrices were different in the settings in the *dXY* columns, where both *X* and *Y* represents a basic concept. This phenomenon is certainly not desirable, though understandable, because of the similarity between *A_B_CD*, *AB_C_D*, and *AB_CD*.

When the heuristics led us to choose a wrong structure, the estimated MI measures for the chosen structures were not larger than the MI measures for the correct structures by a big margin. In fact, we found that, when the heuristics failed to choose the correct answers, most of the estimated MI measures were very close to each other. Hence, we expect that if we consider the ratios between the estimated MI measures, we may design more effective heuristics. We included ratios between estimated measures as features for building the SVM-based classifiers that we report below.

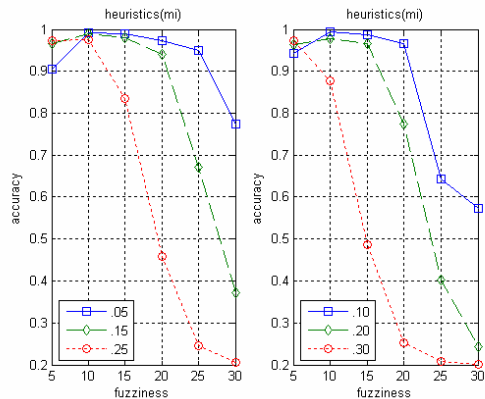


Fig. 4. Accuracy achieved by the MI-based heuristics

4 SVM-Based Methods

Support vector machines [15] are a relatively new formalism that can be applied to the task of classifications. We can train SVMs with training patterns that are associated with known class labels, and the trained SVMs can be used to predict the classes of test patterns. In this work, we employed the LIBSVM packages provided by Chang and Lin [21].

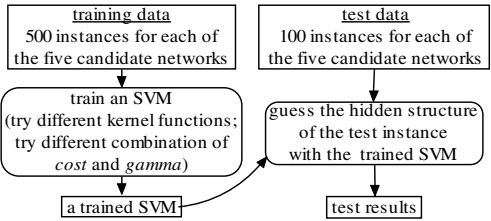


Fig. 5. Guessing the hidden structure with SVMs

4.1 Preparing for Experiments

Figure 5 summarizes the main steps that we took to apply SVMs in our work. As explained in Section 3.2, we obtained students' data in 36 different experiments. In each of these experiments, there were 600 network instances for each of the candidate networks shown in Figure 2. Therefore, we used students' data obtained from 500 network instances for each of the candidate network as the training data, and used the students' data obtained from the remaining 100 network instances as the test data.

Figure 6 summarizes how we prepared the training and test instances. In addition to the original 14 estimated MI measures, we also computed ratios between the estimated MI measures as features. The introduction of ratios was inspired by analyses that we discussed at the end of Section 3.2. We divided the original 14 estimated MI measures by the largest estimated MI measure in each training instance. This gave us 14 new features. We also divided the largest estimated MI measure by the second largest estimated MI measure, and divided the largest estimated MI measure by the average of all estimated MI measures. This gave us 2 more features, so we used 30 features for each of the 500 training instances for each of the five candidate networks. The true answers (also called *class labels*) were attached to the instances for both training and testing. In summary, we created a training instance from 10000 simulated students, and there were 2500 ($=5 \times 500$) training instances, each with 30 attributes and a class label. When testing the trained SVMs, we produced the 16 extra features from the original 14 estimated MI measures for each of the test instances as well. The true answer was attached to the test instance so that we could compare the true and predicted answers, but the SVMs did not peek at the true answers.

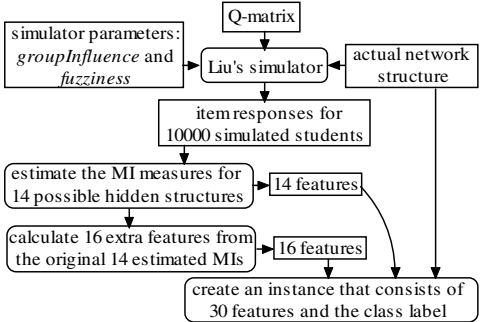


Fig. 6. Preparation of the training and test data

4.2 Results

Charts shown in Figure 7 show the experimental results. The vertical axis, the horizontal axis, and the legend carry the same meanings as those for charts in Figure 4. The titles of the charts indicate what types of SVMs we used in the experiments. We used the c-SVC type of SVMs in all experiments, and tried three different kernel functions, including polynomial (c-svm-poly), radial basis (c-svm-rb), and sigmoid (c-svm-sm) kernels. Among these tests, using polynomial and radial basis kernels gave almost the same accuracy, and both performed better than the sigmoid kernel. However, it took a longer time for us to train an SVM when we used the polynomial kernel.

Comparing the curves for the same experiments in Figures 3 and 4 show the significant improvements achieved by using the SVMs. In the middle chart in Figure 7, the accuracy stays above 0.75 even when *groupInfluence* and *fuzziness* were 0.3. The heuristics-based method got only 0.2 in accuracy under the same situation. In addition, the trends of all curves support our intuitive expectation—larger *groupInfluence* and *fuzziness* would lead to worse accuracy. The problem that occurred in the upper left corners of the charts in Figure 4 was also gone. Even when we tried different Q-matrices, smaller *groupInfluence* and *fuzziness* also made the prediction of the hidden structure easier than when we used larger *groupInfluence* and *fuzziness*.

We have to explain that we had to search for the best parameters for SVMs when we trained SVMs. In particular, we ran experiments that used different values for *cost* and *gamma* in LIBSVM, using default values for other parameters. Different combinations of *cost* and *gamma* led to different accuracy in guessing the hidden structures for the test data. In our experiments we tried combinations of *cost* and *gamma* from values in {0.1, 0.2, ..., 1.9}, and used the best accuracy for the test data in 381 (=19×19) cases when we prepared charts in Figure 7.

5 Concluding Remarks

We tackle a student modeling problem that requires us to infer the hidden model for learning composite concepts, based on observations of variables that have only indirect and uncertain relationships with variables in the hidden model. Experimental results indicate that this task is not impossible, and we can actually achieve good results when the situations are favorable.

We report results of two different approach—A heuristics-based and an SVM-based approach. Charts shown in Figures 3

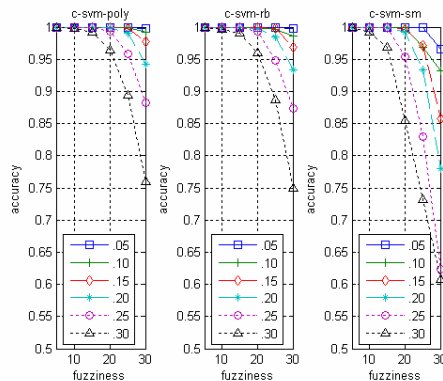


Fig. 7. Accuracy achieved by the SVM-based methods

and 7 clearly show that the SVM-based approach is more effective. However, the advantages of our SVM-based approach come at some costs. We will need experts to enumerate the candidate networks and guess the contents of the Q-matrix. Only after obtaining these information, can we create simulated students and train the SVMs, which will then be used to guess the hidden structure with students' item responses. (Although we did not use item responses of real students in our experiments, we will have to do so in reality.)

Though we cannot discuss all experimental results in this paper, our experience indicates that the contents of the Q-matrix affect the final accuracy. The contents of the Q-matrix show what types of students that we should recruit for investigating how the students learn the composite concepts. Results reported in this paper were acquired with the Q-matrices that assumed students were competent in *dABCD* and all basic concepts. If we allow cells in these columns to be zero, then the final accuracy will be affected. However, we do not think this should happen in reality. If we do want to learn how students learn *dABCD*, we should have tried as hard as possible to collect item responses from students who appear to be competent in *dABCD* and all basic concepts. Given the intentionally introduced uncertainties, i.e., *groupInfluence* and *fuzziness*, our algorithm does allow errors in recruiting students, and experts do not have to provide very exact information about the Q-matrices. The SVM-based approach is reasonably robust in this aspect.

We hope the reported results can be useful for student modeling in reality. Our experience underscores the importance of experts' opinion for the success of the modeling task. Experimental results also show the potential applicability of the heuristics-based methods for selecting the correct hidden sub-structure even when experts' opinions were not available.

Acknowledgements

This research was partially supported by contract NSC-94-2213-E-004-008 of the National Science Council of Taiwan. We gratefully thank the reviewers for their invaluable comments, and will answer their questions that we cannot do so in this page-limited paper during the oral presentation.

References

1. W. J. van der Linden and C. A. W. Glas, *Computerized Adaptive Testing : Theory and Practice*, Kluwer, Dordrecht, Netherlands, 2000.
2. H. Wainer et al., *Computer Adaptive Testing : A Primer*, Lawrence Erlbaum Associates, NJ, USA, 2000.
3. C.-C. Liu, P.-H. Don, and C.-M. Tsai, "Assessment based on linkage patterns in concept maps," *J. of Information Science and Engineering*, vol. 21, pp. 873–890, 2005.
4. G. Smith, "Does Computer-Adaptive Testing Make the Grade?" ABCNEWS.com, 17 March 2003.
5. R. J. Mislevy and R. G. Almond, "Graphical models and computerized adaptive testing," CSE Technical Report 434, CRESST/Educational Testing Services, NJ, USA, 1997.

6. C.-L. Liu, "Using mutual information for adaptive item comparison and student assessment," *J. of Educational Technology & Society*, vol. 8, no. 4, pp. 100–119, 2005.
7. F. B. Baker, *Item Response Theory: Parameter Estimation Techniques*, Marcel Dekker, NY, USA, 1992.
8. R. J. Mislevy, R. G. Almond, D. Yan, and L. S. Steinberg, "Bayes nets in educational assessment: Where do the numbers come from?" in *Proc. of the Fifteenth Conf. on Uncertainty in Artificial Intelligence*, pp. 437–446, 1999.
9. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, CA, USA, 1988.
10. E. Millán and J. L. Pérez-de-la-Cruz, "A Bayesian diagnostic algorithm for student modeling and its evaluation," *User Modeling and User-Adapted Interaction*, vol. 12, no. 2–3, pp. 281–330, 2002.
11. J. Vomlel, "Bayesian networks in educational testing," *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 12, no. Supplement 1, pp. 83–100, 2004.
12. K. VanLehn, Z. Niu, S. Siler, and A. Gertner, "Student modeling from conventional test data: A Bayesian approach without priors," *Lecture Notes in Computer Science*, vol. 1452, pp. 434–443, 1998.
13. D. Heckerman, "A tutorial on learning with Bayesian networks," in M. I. Jordan (ed.), *Learning in Graphical Models*, pp. 301–355, MIT Press, MA, USA, 1999.
14. T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, NY, USA, 1991.
15. C. Cortes and V. Vapnik, "Support-vector network," *Machine Learning*, vol. 20, pp. 273–297, 1995.
16. K. K. Tatsuoaka, "Toward an integration of item-response theory and cognitive error diagnoses," in N. Fredericksen et al. (eds.), *Diagnostic Monitoring of Skill and Knowledge Acquisition*, Erlbaum, NJ, USA, 1990.
17. D. E. Knuth, *The Art of Computer Programming: Fundamental Algorithms*, Addison-Wesley, MA, USA, 1973.
18. P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, second edition, MIT Press, MA, USA, 2000.
19. Hugin: <http://www.hugin.com>
20. I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, second edition, Morgan Kaufmann, CA, USA, 2005.
21. C.-C. Chang and C.-J. Lin, LIBSVM: A library for support vector machines, 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
22. MATLAB: <http://www.mathworks.com>

Practical Approximation of Optimal Multivariate Discretization

Tapio Elomaa¹, Jussi Kujala¹, and Juho Rousu²

¹ Institute of Software Systems, Tampere University of Technology

² Department of Computer Science, University of Helsinki

elomaa@cs.tut.fi, jussi.kujala@tut.fi, juho.rousu@cs.helsinki.fi

Abstract. Discretization of the value range of a numerical feature is a common task in data mining and machine learning. Optimal multivariate discretization is in general computationally intractable. We have proposed approximation algorithms with performance guarantees for training error minimization by axis-parallel hyperplanes. This work studies their efficiency and practicability. We give efficient implementations to both greedy set covering and linear programming approximation of optimal multivariate discretization. We also contrast the algorithms empirically to an efficient heuristic discretization method.

1 Introduction

Much work has been devoted to *univariate* discretization of numerical attribute value ranges [9,10,5,8], but *multivariate* discretization has been left to lesser attention [4,11,1,14,13]. This is because many knowledge discovery and machine learning algorithms cannot take full advantage of multivariate discretizations, even though univariate discretization will unavoidably lose information that depends on several attributes. Even more importantly, optimal univariate discretization takes (after sorting of values) a low-degree polynomial, even linear, and in the best case sub-linear time [9,8], depending on the evaluation function, while optimal multivariate discretization in general is intractable.

The computational complexity forces one to resort to heuristic approaches, which cannot guarantee the quality of the resulting multivariate discretization, or try to develop approximation algorithms with formal performance guarantees for the problem. We have recently proposed approximation schemes for optimal discretization of the Euclidean space using a restricted number of axis-parallel hyperplanes [6]. The aim is not to partition the training sample consistently, which in any case is often impossible and undesirable, but rather to minimize the empirical error. In addition to being an interesting problem in its own right, it also has a connection, e.g., to Naïve Bayes optimal discretization [6].

Our aim in this work is to clarify the practical value of the proposed approximation algorithms in contrast to a commonly-used heuristic algorithm for the same task. We are also interested in the practical efficiency of the approximation algorithms. Though efficient enough in theory, approximation schemes tend to

be relatively time consuming in practice. In this paper we present how they can be implemented efficiently enough for practical purposes.

A training sample $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ of n labeled instances is given in the supervised setting. The instances x_i are composed of the values of d attributes and the class labels y_i come from a small set. We consider the d -dimensional Euclidean instance space $\mathbb{R}^d$. A common subtask of a learning algorithm is to fit a hypothesis closely to the training examples. One often needs to solve an optimization problem of *empirical (or training) error minimization* — finding the hypothesis that errs in classifying as few examples as possible. Fitting the hypothesis too closely to the training sample leads to *overfitting* and, therefore, some form of *regularization* is required. Unfortunately, in many hypothesis classes finding the minimum error hypothesis is computationally intractable.

Value tests for single attributes give naturally rise to axis-parallel hyperplanes. For a numerical attribute A the value test is of the form $\text{val}_A(x) \leq t$, where t is a threshold value. It defines an axis-parallel hyperplane that divides the input space in two half-spaces. We consider the situation where the number of hyperplanes is restricted. If we may choose the number of hyperplanes, we can always guarantee zero error for consistent data by separating each point to its own subset. Minimizing the number of hyperplanes needed to obtain a consistent partitioning is NP-complete, but can be approximated with ratio d in $\mathbb{R}^d$ [2].

The axis-aligned straight lines that a classifier chooses can define a hierarchy of nested half-spaces or they can determine a grid on the whole input space. In, e.g., decision tree learning the root first halves the instance space, its children halve the resulting subspaces, and so forth. The final subspaces (corresponding to the leaves) are eventually assigned with a class label prediction, e.g., by choosing the majority label of the examples in the subspace. In the alternative division the lines always span through the whole input space and penetrate each other.

The grid defined by hyperplanes penetrating each other is a result of executing several attribute value tests simultaneously. Naïve Bayes aims to estimate class densities in the instance space from training data, like any generative classifier, and predicts for each instance the class with the highest density. In the estimation, one applies the (naïve) assumption that the attributes are independent of each other given the class. Together all decision boundaries of the d dimensions divide the input space into a hyper-grid. To fix the labeling of its cells, Naïve Bayes computes the marginal distributions of the attributes and multiplies them, thus implicitly assuming a product distribution.

There are many methods for choosing the hyperplanes to partition the instance space [5]. In unsupervised setting one can divide each dimension into a prespecified number of equal-width or equal-frequency bins. In supervised setting the class values of the training examples are used to guide hyperplane selection. Minimizing empirical error is a natural approach because it gives an upper bound for the accuracy that can be attained using discretized data in contrast to using the original continuous-valued data. Any discretization method which cannot guarantee finding a consistent partition, if one exists, invalidates the possibility of any learning algorithm to work as well as with the original data.

One should, though, not attempt to always partition the training sample consistently, because overfitting will lead to poor prediction accuracy. We will regularize fitting to training data by restricting the number of hyperplanes. Within this class of hypotheses we will search for partitions that are *optimal* in the sense that they minimize empirical error and use as few hyperplanes as possible.

2 Related Work

Univariate supervised discretization techniques only take into account the interaction of the value of one feature and the class value at a time. Hence, they miss dependencies between several variables. These methods are, nevertheless, popular because heuristic approaches are very efficient and produce good result despite missing existing interactions [10,5]. Moreover, many knowledge discovery and machine learning techniques have an inbuilt handling of univariate decisions.

We can sort the data and arrange it into *bins*, each containing all instances with the same value for the attribute (see Fig. 1). Many common evaluation functions cannot distinguish between two adjacent bins that have the same relative class distribution [8]. Hence, we can process the sequence of bins into *segments* by merging together any adjacent bins that have the same relative class distribution. In Fig. 1, for attribute *Y*, four pairs of adjacent bins get merged together. We can discover bins and segments in linear time in one sweep over the data. Hence, the time requirement is dominated by the $O(n \log n)$ time of sorting.

E.g., the straightforward evaluation function empirical error aims at minimizing the number of falsely classified training examples. Each interval is labeled with one of the classes; thus, all instances of the minority classes become errors of the discretization. Two adjacent bins that have the same relative class distribution — and, hence, also the same majority class — can be merged without risking the possibility to come up with an empirical error minimizing discretization. Cut point analysis is not restricted to evaluation functions familiar from decision tree learning, but also extends to naïve Bayesian classifiers [7].

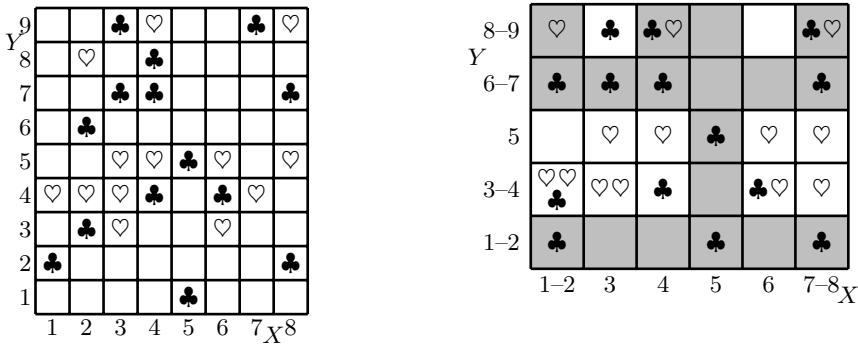


Fig. 1. The bins of two attributes divide the input space into a grid (left). The segments define a coarser grid (right). In the gray area Naïve Bayes predicts class **club**.

When the data in Fig. 1 is classified on the basis of attribute X (resp. Y), eleven (seven) training examples are bound to be misclassified. The same accuracy can be obtained in the data preprocessed into segments. Taking both attributes into account changes the situation. E.g., cell (3,6–7) cannot be determined based on single attributes, because they have different majority classes. Summing up all evidence contained on the relevant row and column (like, e.g., Naïve Bayes does), however, supports predicting **club**. On the other hand, in cell (7–8,8–9) there are equal numbers of instances from both classes on the relevant row and column. One needs to resort to more global properties of the data, e.g., by predicting the more numerous class overall (**club**). Using both attributes to classify the examples, it is possible to attain zero empirical error for the original training data, but the segment division necessitates at least four errors.

The problem of attaining the minimum empirical error is intractable already in two dimensions [3]. Hence, efficient approximation algorithms for this problem are needed. In the following sections two recently proposed algorithms with performance guarantees for empirical error minimization [6] will be reviewed and their practical implementations described.

Despite of the shortcomings of segment-based discretization, it is the way in which Naïve Bayes operates [7]. It predicts the class value by looking at the marginal statistics of all attributes simultaneously because of the assumption of independence given the class. The decision borders of Naïve Bayes fall on segment borders of the data. In Fig. 1 we have denoted by gray the areas in which its prediction would be **club**. Thus, in total seven errors occur. Finding the optimal Naïve Bayesian multivariate discretization is also NP-hard [7].

2.1 Heuristics for Multivariate Discretization

Chmielewski and Grzymala-Busse [4] put forward a multivariate discretization method in which the normalized data is first clustered. The clustering thus obtained is then used to induce a potential discretization of the continuous value ranges by choosing the extreme values of each continuous attribute at each cluster. Finally, adjacent intervals that have the smallest class entropy over all attributes are merged together. In a similar supervised multivariate discretization approach [14] a *neighborhood graph* is first computed for the training sample in cubic time. The edges in this graph depict that connected examples are close to each other in the input space. Then *clusters* of the graph are identified. A cluster is a connected subgraph of the neighborhood graph with vertices (corresponding to training examples) belonging to the same class. Based on the class coherent clusters of the graph discretization is induced by discretizing continuous attribute value ranges at the extreme values of the most relevant clusters.

In the work of Srikant and Agrawal [15] each dimension is first divided into basic intervals and then all combinations of consecutive basic intervals are potential candidates as final intervals. The quadratic number of these combinations per dimension poses a problem that is overcome by computing measures of interestingness for the candidates. Because the minimum support of an enclosing interval is at least the same that of an enclosed interval, one avoids calculating

the values for all intervals. Miller and Yang [12] claimed that the interestingness measures developed for nominal attributes cannot capture the semantics of interval data. Their solution, again, was to first cluster the data and then induce association rules by treating the clusters as frequent item sets.

Also the algorithm of Bay [1] starts by dividing each attribute range finely into intervals using either equal-width or equal-frequency binning. Two adjacent intervals that have minimum combined support and do not already have a discretization boundary in between them are candidates for merging. The multivariate nature of Bay's approach is that a multivariate difference test, rather than an one-dimensional one, is used to compare the similarity of the distributions in the candidate intervals. Intervals with a similar distribution are merged together and otherwise a discretization boundary is placed in between them.

Friedman and Goldszmidt [11] proposed a multivariate discretization technique for Bayesian networks based on the MDL principle, which works either as an unsupervised or a supervised method. It is intertwined with learning of the network structure, which means that only variables adjacent in the network need to be taken into account in choosing the discretization. Another approach tailored for Bayesian networks is that of Monti and Cooper [13]. They assume an underlying discrete mechanism governing the behavior of a continuous variable. This allows specifying a Bayesian scoring metric for discretization policies as the posterior probability of a policy, given the network structure and the data.

3 Approximation Based on Greedy Set Covering

The problem of consistent partitioning of $\mathbb{R}^d$ with axis-parallel hyperplanes $\text{CONSAxis}(S, n)$ is: Given a set S of n points $\{x_1, \dots, x_n\} \subset \mathbb{R}^d$ each labeled either positive or negative, find a consistent partitioning of S with axis-parallel hyperplanes using as few hyperplanes as possible. It seems natural to try to approximate CONSAxis through the problem SET COVER , which is approximable within $\ln n + 1$ [16]. The approximation algorithm attaining the logarithmic ratio is the greedy covering method, which chooses to the evolving cover the subset that contains the largest number of yet uncovered elements [16].

Reducing CONSAxis to SET COVER lets us use the greedy covering algorithm to solve the problem: Given an instance S , we generate a new element corresponding to each conflicting pair of labeled points. Let U be the set of all such elements; in the worst case $|U| = \Omega(n^2)$. It is sufficient to restrict the axis-parallel hyperplanes to a set $\mathcal{H}$ of at most $n - 1$ *canonical hyperplanes* per dimension. There is one such per consecutive points with respect to a dimension. For each we create a set that has a representative for each pair of conflicting examples that can be separated by the hyperplane in question.

The collection of all such sets is an instance of SET COVER such that its solution defines a set of hyperplanes, which separate all conflicting example pairs from each other. Applying now the greedy set covering algorithm gives an approximation for CONSAxis with quality $O(k(1 + 2 \ln n))$, where k is the minimum number of hyperplanes needed to consistently partition S . If the data is

inconsistent, one can remove duplicates of a minority class without losing in attainable accuracy nor having to use an excessive number of hyperplanes.

In practice one often can use only k hyperplanes for the partition. Now the hyperplanes at our disposal do not necessarily suffice to reach the lowest possible error. The goal becomes to attain as low error as possible using them. In the unrestricted case, no polynomial-time algorithm has approximation guarantee $1 + c$ for a constant $c > 0$ [2]. This implies a limitation also to the restricted situation: No polynomial-time algorithm can guarantee an error at most a constant times that of the optimal algorithm using only a constant factor c more hyperplanes.

The main problem of the formulation used above is that it does not give an approximation algorithm for training error. Consider, e.g., four examples divided by a hyperplane into two subsets both containing a positive and a negative example; two conflicts have been removed by the hyperplane, but the error of this partition has not reduced. It might even be possible to reduce the error by separating one, say, positive example from the rest. However, because again only two conflicts are cleared, this choice is not preferred over the first one.

3.1 Pruning Conflicts for Efficient Implementation

A deficiency of set covering approximation is the fact that there can be $\Theta(n^2)$ classification conflicts in a sample of n examples. E.g., the well-known UCI Glass domain only has 214 examples, but almost 20,000 conflicts arise. The quadratic number of conflicts can hamper the efficiency of the otherwise efficient greedy algorithm. It is, however, quite easy to reduce the number of conflicts that need to be taken into account [2]. Consider a pair of examples from different classes and the hyper-rectangle that they span into the instance space. If there is a third example within the rectangle, there is no need to consider this conflict anymore. Any hyperplane that clears the conflict between the third point and one of the corner points will also clear the conflict between the corner points.

Pruning reduces the number of conflicts for UCI Glass down to about 9,000. It is still complicated to efficiently identify conflicting pairs of training examples in the problem reduction. The brute-force implementation would go through all the $O(n^2)$ pairs of examples checking for each conflicting pair in an unintelligent manner whether an example clearing the need to further consider the conflict exists in between them. Altogether this would require $O(n^3)$ time.

Our implementation of conflict pruning is still a cubic method in the worst case, but in practice runs more efficiently. We maintain for each point u a list of conflicts. All other points v are checked for a potential conflict. If the list of u already contains a point allowing pruning of v , it will not be added. Also, we check whether adding v to the list allows to remove some of the previously inserted points. After building this list for u , we delete those items for which $v < u$. This prevents adding a conflicts (u, v) and (v, u) simultaneously. In practice the conflict lists associated with the examples remain quite short.

Experiments have demonstrated that already few hyperplanes suffice to reach consistent partitioning in real-world domains. Hence, in subsequent experiments greedy set covering is run until a consistent discretization has been found.

4 Approximation Based on LP Relaxation

The axis-parallel separation problem formulated as a 0–1 integer program is:

$$\begin{aligned} & \text{minimize } \sum_{i=1}^n w_i \text{ (or, equivalently, } \min \|\mathbf{w}\|_1) \\ & \text{with constraints } \sum_{j=1}^{|\mathcal{H}|} z_j \leq k \text{ and } w_u + w_v + \sum_j z_j \geq 1, \end{aligned}$$

in which a variable w_i , $1 \leq i \leq n$, has value 1 if point i is not separated by the chosen hyperplanes from all points of different class, otherwise $w_i = 0$. The second set of variables z_j represents the canonical hyperplanes. If $h_j \in \mathcal{H}$ is included in the solution, then $z_j = 1$ and otherwise $z_j = 0$.

The first constraint ensures that at most k hyperplanes are chosen. The second one is given for all pairs of points u and v that have different class. In it the sum is taken over all those lines that separate the two points.

The general problem of integer programming is NP-hard, but for a linear program (LP) relaxation many efficient methods for solving are known. In it the variables $\hat{w}_i$ and $\hat{z}_j$ take real values in $[0, 1]$. Let OPT_k be the value of the integer program using k lines and LP_k that of its LP relaxation; $LP_k \leq \text{OPT}_k$. We consider all pairs of points u and v for which, in the fractional solution, it holds $\hat{w}_u + \hat{w}_v \geq C$ for some $0 < C \leq 1$. Rounding up all those variables that have $\hat{w}_i \geq C/2$ ensures that in each such pair at least one variable gets value 1. In the worst case we have to round up $n - 1$ variables. The number of these variables determines the value of the LP relaxation. Hence, this approach can guarantee an approximation ratio of $(2/C) \sum_{i=1}^n \hat{w}_i = 2 LP_k / C \leq 2 \text{OPT}_k / C$.

Any two points u and v , that still need to be separated now have $\hat{w}_u + \hat{w}_v < C$. Because of the second constraint, in the solution of the LP relaxation $\sum \hat{z}_j \geq 1 - C$. In order to include one of the relevant hyperplanes to our solution we can go through the hyperplanes dimension by dimension and cumulate the sum of their $\hat{z}$ values. In the plane as long as the sum is below $(1 - C)/2$ we round the $\hat{z}$ variables of the lines down to 0. We choose all those lines that make the total sum reach $(1 - C)/2$. The sum is then reset to 0 and we continue to go through the lines. As the sum of the fractional variables also obeys the upper bound of k , this way we may end up picking at most $2k/(1 - C)$ lines to our approximation.

Let APP_k be the value of the rounding procedure and line selection using k lines. In the plane $\text{APP}_{2k/(1-C)} \leq 2 \text{OPT}_k / C$. As necessitated, this algorithm uses more lines and makes more false classifications than the optimal solution. E.g., when $C = 1/2$, we have $\text{APP}_{4k} \leq 4 \text{OPT}_k$. The general form of the performance guarantee in d dimensions is $\text{APP}_{dk/(1-C)} \leq 2 \cdot \text{OPT}_k / C$.

4.1 Dual Formulation and Other Implementation Details

Consider the unrestricted problem in which our objective is to minimize the number of chosen hyperplanes; i.e., $\min \mathbf{z}$, where $\mathbf{z} = \sum_{j \in \mathcal{H}} z_j$. The lines chosen should resolve all classification conflicts. I.e., we want to have for each conflict $\sum_j z_j \geq 1$, where the sum is now taken over the hyperplanes that separate the two points in conflict. Let us gather these conditions to the rows of a matrix A .

A feasible solution must satisfy the constraint $A\mathbf{z} \geq \mathbf{1}$. Altogether, we now have an integer program which minimizes $\|\mathbf{z}\|_1$ with constraints $A\mathbf{z} \geq \mathbf{1}$ and $\mathbf{z} \geq 0$. In solving the problem with a restricted number of hyperplanes, we have to include the variables representing the points into $\mathbf{z}$ and A , and modify the function to be minimized. Moreover, A has to include the constraint $\sum_{j \in \mathcal{H}} z_j \leq k$. The LP-solver that we use (Matlab's LIPSOL) works clearly more efficiently for the dual formulation of the problem. Directly by definitions we get the dual formulation for our primal problem: maximize $\|\mathbf{u}\|_1$ with constraints $A^T \mathbf{u} \leq 1$ and $\mathbf{u} \geq 0$.

A straightforward implementation of the LP needs a matrix of size $\gamma(n + |\mathcal{H}|)$, where γ is the number of conflicting example pairs. Both the number of conflicts and hyperplanes to consider can be pruned. To reduce γ one can use the pruning of points explained above [2] and to bring the number of hyperplanes down, one can leave those in between two points of the same class without any attention. It can happen that even after pruning $\gamma = \Theta(n^2)$, but in a good case $\gamma = \Theta(n)$.

LP-solvers work, e.g., in time $O(d^3 l^2)$, where l is the length of the input in bits. Thus, the running time does not depend on the number of constraints. Because each constraint only gives two adjacent sets of lines, we can code the input shortly. Moreover, this LP can be presented using a sparse matrix of constraints with only a fraction of about $2d/n$ non-zero elements. The new matrix is composed of variables σ_ℓ each of which is a cumulative sum over lines in the direction of one axis. In the plane the sum of z_j over the lines that separate u and v can now be computed by summing the σ -variables which correspond to the lines in between the points with the highest x -coordinate and y -coordinate values. From this sum we subtract the σ -variables that precede the first line in between the two points within the direction of both coordinate axes. Each constraint, thus, has only $2d$ variables independent of the number of points or hyperplanes. We can, therefore, employ sparse matrix techniques to speed up the linear program.

Efficiency is also a major concern in using the LP-approximation, despite the theoretical efficiency of LP-solvers. The pruning explained in the previous section makes it possible to use the approach, but it still leaves a lot of computational burden. The approximately 20,000 potential conflicts of the UCI Glass domain, e.g., drop down to about 9,000 ($c d n$, where c is some constant) after pruning, but still lead to a LP with some 200,000 ($2d \cdot \gamma$) non-zero entries. Anyway, this is an improvement of an order of magnitude, because without using the z_j variables, the number of non-zero entries would have been around 2,000,000.

5 Empirical Evaluation

We now test the quality of the approximations. As the test bench we use Naïve Bayes. The reference discretization methods is equal-width binning, which is a commonly employed technique (e.g., [5]). Laplace correction is applied in class prediction in all our algorithms. Let $|B_i|$ be the number of examples in cell B_i of the chosen discretization and $|B_i^c|$ the number of those that are instances of class $c \in C$. The fraction of instances of class c is counted as $(|B_i^c| + 1)/(|B_i| + |C|)$.

The **LP** relaxation of the integer program has parameter settings $k = 5$ and $C = 0.25$. This restriction on the number of hyperplanes guarantees that the

number of hyperplanes after rounding is linear in d . We use the dual formulation and Matlab’s LIPSOL package for solving LPs by interior point methods. **Greedy** set covering includes preprocessing of conflicts and it is run until a consistent solution is found. The results show that there is no need to restrict the number of hyperplanes. In **EqWidth** each dimension of the domain is discretized into ten equally wide bins in between the smallest and the largest observed value.

Table 1. Prediction accuracies on training and test sets with standard deviations

Domain	LP			Greedy			EqWidth	
	Train	Test	cells	Train	Test	cells	Train	Test
Australian	85.0 $\pm$ 1.2	86.9 $\pm$ 2.5	19.6	86.2 $\pm$ 1.3	86.9 $\pm$ 2.9	29.2	86.1 $\pm$ 1.2	86.5 $\pm$ 3.0
Breast W.	97.8 $\pm$ 0.5	96.8 $\pm$ 1.0	32.1	97.3 $\pm$ 0.7	96.0 $\pm$ 0.7	18.6	97.8 $\pm$ 0.5	97.1 $\pm$ 0.7
Diabetes	76.9 $\pm$ 0.5	76.0 $\pm$ 2.1	44.8	76.9 $\pm$ 0.8	74.6 $\pm$ 2.0	27.4	77.5 $\pm$ 0.3	77.7 $\pm$ 1.7
Ecoli	92.2 $\pm$ 2.8	84.5 $\pm$ 4.0	25.7	89.5 $\pm$ 3.7	84.0 $\pm$ 4.9	20.9	89.7 $\pm$ 2.6	84.2 $\pm$ 5.3
German	74.8 $\pm$ 1.4	72.5 $\pm$ 2.2	39.3	75.4 $\pm$ 1.6	72.9 $\pm$ 1.6	40.8	75.6 $\pm$ 1.0	72.1 $\pm$ 1.7
Glass	79.9 $\pm$ 4.2	64.1 $\pm$ 5.7	44.8	71.9 $\pm$ 3.0	60.8 $\pm$ 5.2	22.3	74.9 $\pm$ 3.0	56.1 $\pm$ 7.4
Ionosph.	93.2 $\pm$ 1.2	89.3 $\pm$ 3.0	76.9	91.2 $\pm$ 1.7	87.8 $\pm$ 3.2	41.9	89.6 $\pm$ 1.4	85.4 $\pm$ 1.8
Iris	96.2 $\pm$ 1.8	93.9 $\pm$ 4.0	13.2	95.7 $\pm$ 1.7	95.9 $\pm$ 2.8	10.1	96.0 $\pm$ 0.8	95.1 $\pm$ 2.1
Liver	75.0 $\pm$ 3.3	63.7 $\pm$ 4.9	31.5	72.2 $\pm$ 2.8	63.5 $\pm$ 2.1	22.4	70.4 $\pm$ 3.1	59.9 $\pm$ 5.3
Segment.	95.0 $\pm$ 2.4	80.3 $\pm$ 3.6	66.1	86.2 $\pm$ 3.0	77.7 $\pm$ 5.7	28.5	90.4 $\pm$ 2.3	78.1 $\pm$ 4.5
Sonar	84.1 $\pm$ 3.2	73.9 $\pm$ 3.7	141.0	78.0 $\pm$ 3.1	73.7 $\pm$ 5.8	66.7	95.2 $\pm$ 1.8	76.5 $\pm$ 3.5
Wine	97.8 $\pm$ 1.3	93.2 $\pm$ 3.1	43.3	97.9 $\pm$ 1.7	91.7 $\pm$ 3.9	18.7	98.6 $\pm$ 0.6	96.8 $\pm$ 1.7
	LP $\leftrightarrow$ Greedy			LP $\leftrightarrow$ EqWidth			Greedy $\leftrightarrow$ EqWidth	
Significant	6 $\leftrightarrow$ 1	1 $\leftrightarrow$ 0		5 $\leftrightarrow$ 4	2 $\leftrightarrow$ 3		0 $\leftrightarrow$ 3	1 $\leftrightarrow$ 3

We test our algorithms on well-known domains from the UCI repository. Table 1 lists the average training and test accuracies obtained over ten independent repetitions. As training data a randomly chosen portion of 2/3 of the examples was used and the remaining 1/3 was reserved for testing. Moreover, the average numbers of cells in the chosen partition are listed (for EqWidth this number is $10d$.) The bold figures indicate the best training and test accuracies, respectively.

LP clearly gives the best training accuracies. It is also most often the most accurate classifier on test data. Hence, the quality of optimization obtained through linear programming is high. The number of cells in the approximation of LP is almost always clearly higher than in that of Greedy. However, this is not reflected in the prediction accuracies as overfitting. Greedy is surprisingly weak in training error; it is usually worse than the simple EqWidth method on training data, but fares better on the test data.

Table 1 also summarizes a pairwise comparison of statistically significant differences obtained in both training and test accuracies. The figures denote the numbers of differences deemed as statistically significant by Student’s t -test at 95% confidence level. In light of these tests LP and EqWidth fare equally well and Greedy cannot compete with them. Many statistically significant differences are obtained in training accuracy, but on unseen data the differences level down.

6 Conclusion

Multivariate discretization has been used in practice only in the limited scale even though it is based on using more of the available information than univariate discretization. One of the problems of multivariate discretization is efficiency. For example, optimal multivariate discretization of the Euclidean space using axis-parallel hyperplanes studied in this paper is NP-complete. Therefore, we have developed approximation algorithms with performance guarantees for the problem. Our aim in this work was to study the practicality of the proposed approximation schemes. As a future work we study possible regularization schemes for the number hyperplanes in the relaxation of the problem.

References

1. S.D. Bay: Multivariate discretization of continuous variables for set mining. In: Proc. 6th ACM SIGKDD. ACM Press (2000) 315–319
2. G. Călinescu, A. Dumitrescu, P.-J. Wan: Separating points by axis-parallel lines. In Proc. 16th Canadian Conference on Computational Geometry. (2004) 7–10
3. B.S. Chlebus, S.H. Nguyen: On finding optimal discretizations for two attributes. In: Rough Sets and Current Trends in Computing. Volume 1424 of LNAI. Springer (1998) 537–544
4. M.R. Chmielewski, J.W. Grzymala-Busse, Global discretization of continuous attributes as preprocessing for machine learning. *Int. J. Approximate Reasoning*, 15 (1996) 319–331.
5. J. Dougherty, R. Kohavi, M. Sahami: Supervised and unsupervised discretization of continuous features. In: Proc. 12th ICML. Morgan Kaufmann (1995) 194–202
6. T. Elomaa, J. Kujala, J. Rousu: Approximation algorithms for minimizing empirical error by axis-parallel hyperplanes, In: Proc. 16th ECML. Volume 3720 of LNAI. Springer (2005) 547–555
7. T. Elomaa, J. Rousu: On decision boundaries of Naïve Bayes in continuous domains. In: Proc. 7th PKDD. Volume 2838 of LNAI. Springer (2003) 144–155
8. T. Elomaa, J. Rousu: Efficient multisplitting revisited: Optima-preserving elimination of partition candidates, *Data Mining and Knowl. Discovery*, 8 (2004) 97–126.
9. U.M. Fayyad, K.B. Irani: On the handling of continuous-valued attributes in decision tree generation. *Machine Learn.*, 8 (1992) 87–102.
10. U.M. Fayyad, K.B. Irani: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proc. 13th IJCAI. Morgan Kaufmann (1993) 1022–1027
11. N. Friedman, M. Goldszmidt: Discretizing continuous attributes while learning Bayesian networks. In: Proc. 13th ICML. Morgan Kaufmann (1996) 157–165
12. R.J. Miller, Y. Yang: Association rules over interval data. In: Proc. 1997 ACM SIGMOD. ACM Press (1997) 452–461
13. S. Monti, G.F. Cooper: A multivariate discretization method for learning Bayesian networks from mixed data. In: Proc. 14th UAI. Morgan Kaufmann (2002) 404–413
14. F. Muhlenbach, R. Rakotomalala: Multivariate supervised discretization, a neighborhood graph approach. In: Proc. 2002 ICDM. IEEE CS Press (2002) 314–321
15. R. Srikant, R. Agrawal: Mining quantitative association rules in large relational tables. Proc. 1996 ACM SIGMOD. ACM Press (1996) 1–12
16. V.V. Vazirani: *Approximation Algorithms*. Springer, Berlin, Heidelberg (2001)

Optimisation and Evaluation of Random Forests for Imbalanced Datasets

Julien Thomas^{1,2}, Pierre-Emmanuel Jouve², and Nicolas Nicoloyannis¹

¹ Laboratoire ERIC, Université Lumière Lyon2, France

² Company Fenics Lyon, France

Abstract. This paper deals with an optimization of Random Forests which aims at: adapting the concept of forest for learning imbalanced data as well as taking into account user's wishes as far as recall and precision rates are concerned. We propose to adapt Random Forest on two levels. First of all, during the forest creation thanks to the use of asymmetric entropy measure associated to specific leaf class assignation rules. Then, during the voting step, by using an alternative strategy to the classical majority voting strategy. The automation of this second step requires a specific methodology for results quality assessment. This methodology allows the user to define his wishes concerning (1) recall and precision rates for each class of the concept to learn, and, (2) the importance he wants to confer to each one of those classes. Finally, results of experimental evaluations are presented.

1 Introduction

Imbalanced data learning constitutes one of the greatest challenges for machine learning and KDD communities [1,2]. Learning imbalanced data may occur in a large number of situations such as identification of fraudulent credit card transactions [3], pre-term birth prediction [4], telecommunication equipments failures detection [5], and many more. Several approaches have been proposed to deal with this problem. They may be classified into two distinct classes of methodologies: (1) cost sensitive methods which consider that prediction errors on certain classes of the concept to learn are judged as more penalizing by the model [6,7] ; (2) sampling based strategies (oversampling, downsampling and stratified sampling) which aims at adding/removing instances from the learning sample in order to get a "more balanced" dataset (like Smote [8]).

Our proposal is to adapt the classical Random Forest model [9]. Some adaptations of Random Forests have already been proposed by Chen & Liaw [10] for imbalanced data. However, they were based on previously introduced cost and sampling based methodologies. Here, we consider adapting Random Forest with different orientations and strategies in order to get learning results which better correspond to user's expectations, particularly concerning user's expectations about precision and recall rates.

2 Random Forests Adaptation

A Random Forest (RF) [9] is an ensemble of classification trees, each of them being grown to the largest extent on a bootstrap of the learning sample. Furthermore, for each (node) split step during the growing process, a subsample of k variables of the whole set of variables of the learning problem is drawn at random and the best split on these k variables is used. (Note that the value of k is held constant during the forest growing). Classification of an object is obtained by getting the classification of each tree (we say the tree votes for a class) and then choosing the classification having the most votes over all the trees of the forest. The voting strategy is clearly a majority vote strategy. RFs' performances are usually significantly superior to those of a single classification tree such as C4.5 [11]. RFs are also more robust and present better generalisation ability [9].

However, RFs' parameters (trees and variables numbers) do not allow to reach in a direct way our goals which are: to get better results on imbalanced data and to allow the user to specify his preferences in terms of recall and precision rates on each class of the concept to learn.

We think that modifications may be done on two different levels:

During Each Tree Growing

- Using another entropy measure (Gini index is classically used) in order to redefine the preferences between distributions and consequently to adapt these preferences to imbalanced data.

- Using other class assignation rules for leaves. Usually the most represented class of a leaf is always assigned to the considered leaf. It's possible to use more subtle rules based on the composition of the leaf. This may help in (1) increasing recall over marginal classes, (2) introducing a kind of priority between classes and increasing recall of priority classes.

- Limiting tree depth in order not to get "pure" (one class) leaves. Thanks to a combined use of specific class assignation rules, this may help in increasing recall rate on certain classes.

- Modifying the number k of variables to consider at each splitting step, in order to get more or less independent trees.

During Trees Votes Aggregation

- Majority vote strategy may be replaced by other strategies which may involve costs or weightings in order to readjust recall and precision rates on specified classes of the concept to learn.

As for the tree growing process, we propose here to modify: (1) the entropy measure in order to adapt this key element to the imbalance of datasets ; (2) the leaf class assignation rules in order to increase the recall rates for minority classes by making them priority.

Concerning the voting strategy, we propose to tune it to the modifications applied on the tree growing process and consequently recovering a high precision rate for the forest. Those modifications (which basically consist in an automated

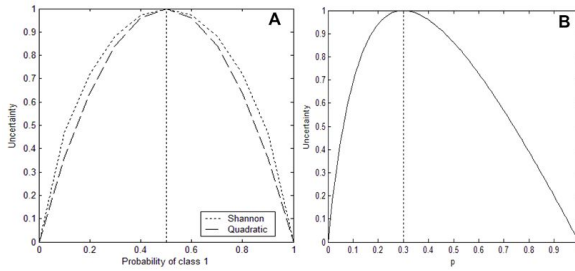


Fig. 1. Classical(A) vs asymmetric(B) entropy measures for 2 classes problem [12]

vote weighting) imply enhancement on global RFs' performances on imbalanced data. Furthermore, user's expectations concerning recall and precision rates on each class are also taken into account: the automated vote weighting procedure consists in the optimization of a criterion which models user's preferences between recall and precision rates on each class of the concept to learn. We now detail our proposed modifications.

Entropy Measure. Classical entropy measures are not well suited to imbalanced datasets due to their symmetry (Fig. 1). Actually, such measures tend to prefer distributions which are far from a balanced distribution. Yet, in case of imbalanced datasets, we do not think that the objective should be to get as far as possible from the balanced distribution, but that it should be to get as far as possible from the initial (imbalanced) data distribution. This stands for the reason why we decide to use an entropy measure that takes into account the imbalance by redefining what should be "good distributions". An asymmetric entropy measure (Fig. 1) [12] allows to take into account those elements.

Class Assignment Rules. Tree leaves obtained with an asymmetric entropy measure may be such that the minority classes are more frequent than they are in the initial learning set, but this frequency increase is often not enough to make them the majority. Consequently, we design some class assignment rules different of the majority assignment. To do so, we define for each class a frequency threshold. If, for a leaf, none of the classes reaches this threshold, objects of the leaf will be considered as not classified by the tree (i.e. the tree does not vote for these object). If one and only one class reaches its threshold then this class is chosen by the tree for the objects of the considered leaf (i.e. the tree votes this class for these objects). Finally, if several classes reach their threshold, the finally chosen class is the one having the lower threshold (this allows the user to give some kind of priority on certain classes over others). To avoid problems due to equality we decided to prohibit equal thresholds. This implies a kind of strict order on preference on classes.

Formally, consider a concept to learn with n classes $(M_1, \dots, M_n)$ and a leaf which has the following frequency distribution $F = \{f_1, \dots, f_i, \dots, f_j, \dots, f_n\}$. User defines an ensemble of frequency thresholds

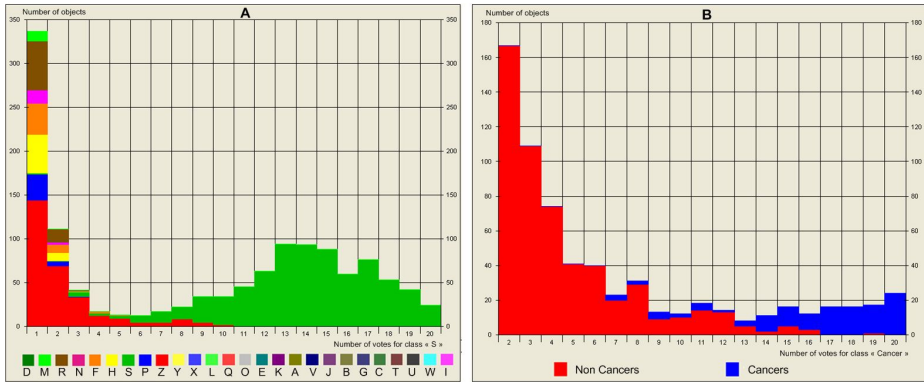


Fig. 2. (A) Example of votes distributions on: (A) Letters dataset [15] (objects without any votes for S class are not represented) ; (B) Mammo dataset (see section 4) (objects with less than 2 votes for Cancer class are not represented)

$$S = \{s_1, \dots, s_i, \dots, s_j, \dots, s_n\} \mid \forall i \neq j, s_i \neq s_j$$

Let I be the ensemble of assignable modalities

$$I = \{i \in \{1, \dots, n\} \mid f_i > s_i\}$$

The class assignation rule for a leaf is:

If I is empty, the objects of this leaf are not classified, else the assigned class is M_i if $i \in I$ and $\forall j \in (I - \{i\}), s_i < s_j$

Voting Strategy. Our voting strategy is designed to give more or less importance to votes attributed by trees (Fig. 2). User determines a weight for each class, which multiplies the number of votes get by the object for this class. Thus, the assigned class for an object is not always the class for which it receives the more votes but the class for which the number of votes multiplied by its weight is the greatest. This allows first to increase the recall rates on minority classes by assigning them strong weights. Then, this only implies to determine one parameter (weight) per class, which is quite simpler than cost based approaches which need defining one parameter per couple of classes.

3 Automatic Votes Weightings by Model Assessment

It might be quite hard to find manually the weightings providing the best fit between forest results and user's expectations. When the concept to learn have only two classes this may be possible (this may be seen as a kind of frontier shift in term of number of votes). However, when the concept to learn have more than two classes the number of parameters settings (ratios between each couple of weightings) becomes far greater and it is consequently obvious that the weightings discovery must be automatic.

Models Assessment. To find "good" weightings, we define a criterion to evaluate model results according to user's expectations. Classically, model optimization is done through global error rate minimization; yet, this is not well-adapted to strongly imbalanced datasets since in this case the importance of a class is proportional to its number of objects. Another drawback is to not allow to use votes weightings for tuning classes recall and precision rates because no distinction is made between error types. To avoid those two major drawbacks we propose a new way to evaluate models called PRAGMA (Precision and RecAll Guided Model Assessment).

PRAGMA is based upon two principles: the notion of importance of a class and the notion of preference between recall rate and precision rate for each class.

First of all, the importance of a class is represented by a coefficient θ_i set by the user. The importance is used at the end of the model evaluation.

Then, for each class, the model is evaluated according to its recall(r_i) and precision (p_i) rates. This function $f(r_i, p_i)$ (that has to be minimized in analogy with the global number of errors for a model) must share the following properties:

- (1) $f(0, 0) = 1$, this corresponds to the worst situation ($r_i = 0$ and $p_i = 0$)
- (2) $f(1, 1) = 0$, this corresponds to the best situation ($r_i = 1$ and $p_i = 1$)
- (3) $\frac{df(r, p)}{dr} < 0, r \in [0; 1]$, if two situations have the same precision rate then the situation with the higher recall rate is considered as the best one.
- (4) $\frac{df(r, p)}{dp} < 0, p \in [0; 1]$, if two situations have the same recall rate then the situation with the higher precision rate is considered as the best one.

Such a function may have the following form: $f(r_i, p_i) = 1 - 0.5(r_i + p_i)$

In order to take into account user's expectations in term of preference between recall rate and precision rate we decided to weight both r_i and p_i :

$$f(r_i, p_i) = 1 - 0.5(\lambda \times r_i + \Omega \times p_i) = 1 + \alpha \times r_i + \beta \times p_i$$

Indeed the ratio α/β determines the preference between recall and precision (the higher it is (above 1), the more recall is preferred ; the lower it is (below 1), the more precision is preferred ; if it is equal to 1, it means that no distinction is made between recall and precision).

To set in an understandable way those 2 parameters, the user has to provide two extremes situations in term of recall and precision which he judges as equivalent. Practically, these situations are: the situation with a perfect recall rate ($r_i = 1$), and the situation with a perfect precision rate ($p_i = 1$). Consequently, it implies that the user sets 2 values x and y (x stands for a recall rate and y for a precision rate) such that $f(1, x) = f(y, 1)$. Setting those 2 values may be seen as answering the following two questions:

- Which compromise on precision do you agree to make to get a perfect recall rate ($r_i = 1$)? (answering this question permits to set x , with $0 \leq x < 1$)
- Which compromise on recall do you agree to make to get a perfect precision rate ($p_i = 1$)? (answering this question permits to set y , with $0 \leq y < 1$)

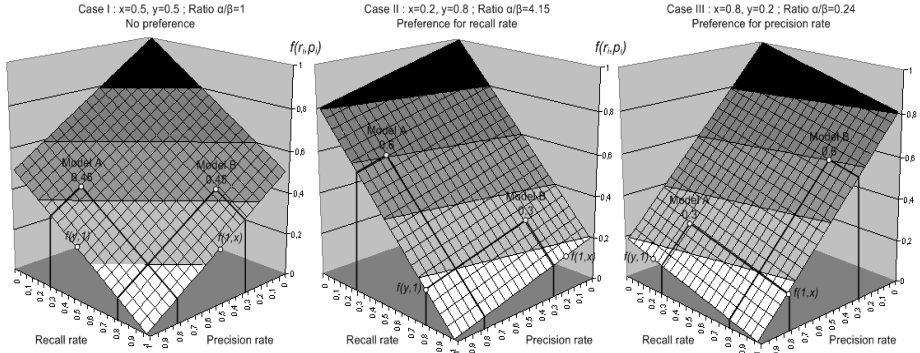


Fig. 3. 2 models, A($r = 0.3; p = 0.8$) and B($r = 0.8; p = 0.3$), are evaluated with 3 different $f(r_i, p_i)$: Case I (symetric) A et B are equivalent; Case II (recall preferred) B is better; Case III (precision preferred) A is better

With this last constraint ($f(1, x) = f(y, 1)$), we can determine both α and β parameters:

$$\alpha = \frac{-1}{1 + \frac{(1-y)}{(1-x)}} \text{ and } \beta = \frac{1}{1 + \frac{(1-y)}{(1-x)}} - 1$$

So, the function used to evaluate a model according to one class's precision and recall rate is the following:

$$f(r_i, p_i) = \frac{-1}{1 + \frac{(1-y)}{(1-x)}} \times r_i + \left(\frac{1}{1 + \frac{(1-y)}{(1-x)}} - 1 \right) \times p_i + 1$$

Note that this corresponds to the equation of a plan where the axe defined by points (0,0,1) and (1,1,0) is fixed and where the ratio α/β determines the orientation of this plan around this axe (the preference between recall and precision rate)(Fig. 3).

These local evaluations (evaluations according to each class) are subsequently combined through a weighted average for which importance coefficients are weights: $PRAGMA = \frac{1}{\sum_{i=1}^n \theta_i} \sum_{i=1}^n \theta_i \times f_i(r_i, p_i)$, with n =number of classes.

Weightings Automatic Discovery. The algorithm used here for the votes weightings automatic discovery is a simulated annealing based algorithm [14]. It is well suited and efficient to solve this optimization problem. The extra computational cost (compared to classical Random Forests) is quite low. The RF is built only once and the unique difference with classical Random Forests lies in the fact that there are several voting steps whereas there is only one in classical RFs, each voting step correspond to a set of votes weightings and to one optimization step. In fact, for each votes weightings set, the classification results has to be updated in order to get a PRAGMA evaluation of the model and consequently guide the optimization process. However, by initially storing for each object the number of non weighted votes that it has received for each class, it is very easy and quite fast to re-compute the final classification obtained with votes weightings.

4 Experimental Evaluations

This section is dedicated to the presentation of results obtained on 3 imbalanced datasets. The first two ones, satimage and hypothyroid, are reference datasets [15]. For those two datasets, the number of classes has been reduced to two (as in [10]) by conserving the minority class and fusing together all other classes in order to compare our results to those presented in Chen and Liaw [10]. The last dataset, Mammo, comes from an industrial medical application for breast cancer diagnosis. Compositions of these datasets are given in Table 1.

Table 2 shows results obtained for a 10-folds cross validation. Parameters settings were chosen in order to give a preference to the recall rate of the minority class (Parameters: Number of trees per RF was set to 20 for each experiment; numbers of variables for randomization was set to 6, 5 and 15 for respectively satimage, hypothyroid and mammo datasets; concerning leaf class assignation rules parameters were set to 0.8 and 0.2 for respectively majority and minority class; finally, for PRAGMA evaluation, importance coefficients were : 1 for the majority class and 10 for the minority class, couples of x and y values were 0.5, 0.5 for the majority class and 0.1, 0.9 for the minority class). Algorithm C4.5 is used there as a kind of reference to illustrate difficulties implied by each one of these learning problems.

Two major observations may be done:

- PRAGMA seems to efficiently translate the user's wish to favour the recall rate of the minority class.
- Simultaneous use of Random Forests with an asymmetric entropy measure as well as with votes weightings and their optimization thanks to PRAGMA method always give the best results according to the user's objectives.

Detailed results obtained with a 10 folds cross-validation on mammo dataset allow a better description of votes weightings effects and PRAGMA method effects on performances of a modified Random Forest (A Random Forest using asymmetric entropy, composed by 20 trees, considering 10 variables for randomization, and using votes weighting without their optimization thanks to PRAGMA method). Four indices (recall rate, precision rate, overall error number, pragma evaluation) are measured for different values of the ratio R : Cancer class weighting / Non Cancer class weighting (Fig. 4).

We remark that the stronger variations for recall and precision rates occur for Cancer class which is due to its low number of objects. Concerning the overall number of error two points are noticeable: (1) its graph is asymmetric which is explained by the fact that a low variation of the majority (Non Cancer) class

Table 1. Datasets description

Dataset	Objects	Features	Frequency of minority class (%)
Satimage	6435	36	9.73
Hypothyroid	3772	27	7.71
Mammo	3528	134	5.00

Table 2. Results for Satimage, Hypothyroid and Mammo datasets

Satimage Methods	Class 4		Overall
	Recall (%)	Precision (%)	correct rate (%)
C4.5	55.0	59.0	91.9
Classical RF	50.9	83.2	94.2
BRF [10]	77.0	56.0	91.9
WRF [10]	77.0	61.0	93.0
Asymetric RF	76.3	54.4	91.4
Asymetric RF optimised	80.7	53.6	91.4
Hypothyroid Methods	Class Negative		Overall
	Recall (%)	Precision (%)	correct rate (%)
C4.5	96.9	95.2	99.4
Classical RF	95.1	94.2	99.2
BRF [10]	95.0	63.0	95.3
WRF [10]	93.0	83.0	98.0
Asymetric RF	94.8	95.5	99.3
Asymetric RF optimised	98.2	94.3	99.4
Mammo Methods	Class Cancer		Overall
	Recall (%)	Precision (%)	correct rate (%)
C4.5	8.3	66.7	98.1
Classical RF	29.2	84.5	98.5
Classical RF optimised	43.5	82.0	98.7
Asymetric RF	33.3	71.8	98.4
Asymetric RF optimised	64.9	50.3	98.1

recall rate, due to a strong weighting for minority class, implies logically more errors than a low variation of the minority (Cancer) class recall rate;(2) for ratios $2 \leq R \leq 5$ there is a very slight variation in the overall number of errors whereas the nature of errors varies a lot (see recall and precision rates graphs). It seems that a kind of transfer of errors occurs:

$R \leq 2$: misclassified object are in majority objects of the Cancer class

$3 \leq R \leq 4$: balanced misclassification (proportion of Cancer and Non Cancer misclassified objects are the same)

$5 \leq R$: misclassified object are in majority objects of the Non Cancer class

PRAGMA measure implies different observations. First, the asymmetry is inverted, which shows that variations of the Cancer recall rate have a strong effect on the measure (this is due to the high importance coefficient of Cancer class as well as to the recall rate preference due to the parameters settings). Then, for ratios $2 \leq R \leq 5$, variations are far more important than for corresponding values of the overall number of errors. This stands for the fact that the nature of errors is taken into account by PRAGMA measure for which misclassified Cancer objects have a greatest impact than misclassified non Cancer objects. It clearly appears that user's wishes to favour Cancer class objects classification and to prioritize Cancer class recall rate are taken into account by PRAGMA measure.

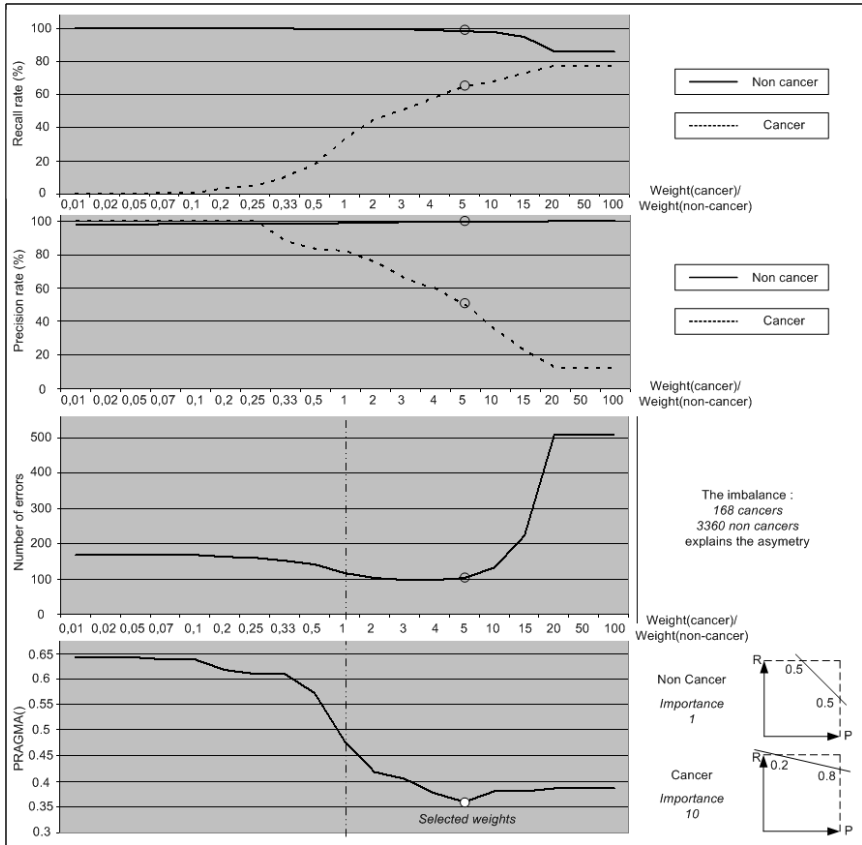


Fig. 4. Detailed results for Mammo dataset

5 Conclusion

We propose a modification of RF for imbalanced datasets that allows the user to specify its expectations in terms of recall and precision rates for each class of the concept to learn. This modification is a two level adaptation of RF:

(1) Trees Growing Level:

- replacing the classical entropy measure by an asymmetric entropy measure
- using special leaf class-assignment rules

(2) Voting strategy level:

- using a classes based votes weighting strategy
- optimizing the weightings thanks to PRAGMA model assessment method

This modifications imply only a low extra computational cost but yields to models enhancement and to the possibility to specify wishes in terms of preference between recall and precision rates for each class of the concept to learn.

We plan to experiment other functions for evaluating models according to one class: we use here a function equivalent to the equation of plan but we can fancy

using different kind of functions or slightly modifying our modelisation of user's preference between recall and precision rates which would imply to use another type of function to describe our modelisation. We also plan to experiment the use PRAGMA measure for building trees instead of using classical entropy measures.

Acknowledgement. We would like to thank all members of Fenics Company especially S. Marcellin, J. Clech, A.S. Darnand, as well as E. Prudhomme and G. Legrand from University Lyon2 for their help and advices. Support for this work was partially provided by the French Ministry for Research and Industry.

References

1. G.M. Weiss, "Mining with Rarity: A Unifying Framework", SIGKDD Explorations, 6(1):7-19, 2004.
2. N. Japkowicz, "The Class Imbalance Problem: Significance and Strategies", Proc. of International Conf. on Artificial Intelligence (IC-AI'2000), 2000.
3. P.K. Chan, S.J. Stolfo, "Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection." Proc. of the 4th International Conf. on Knowledge Discovery and DataMining, 164-168, 2001.
4. J.W. Grzymala-Busse, Z. Zheng, L. K. Goodwin, and W. J. Grzymala-Busse, "An approach to imbalanced data sets based on changing rule strength." Learning from Imbalanced Data Sets: Papers from the AAAI Workshop, pp 69-74, AAAI Press Technical Report WS-00-05, 2000.
5. G.M. Weiss, and H. Hirsh, "Learning to predict rare events in event sequences." Proc. of the 4th International Conference on Knowledge Discovery and Data Mining, pp 359-363, 1998.
6. P. Domingos, "Metacost: A general method for making classifiers cost-sensitive.", Proc. of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 155-164, San Diego, ACM Press, 1999.
7. Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., Brunk, C., "Reducing misclassification costs.", Proc. of the 11th International Conference on Machine Learning, San Francisco. Morgan Kaufmann, 1994.
8. Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P. "Smote: Synthetic minority oversampling technique.", Journal of Artificial Intelligence Research, 16, pp 321-357, 2002.
9. L. Breiman, "Random Forests", In Machine Learning, vol 45(1), pp 5-32, 2001.
10. C. Chen, A. Liaw, "Using Random Forest to Learn Imbalanced Data.", Technical Report. Berkeley, Department of Statistics, University of California, 2004.
11. F. Leon, M.H. Zaharia, D. Gálea, "Performance Analysis of Categorization Algorithms", Proc. of the 8th International Symposium on Automatic Control and Computer Science, ISBN 973-621-086-3, 2004.
12. S. Marcellin, D.A. Ziguéd, G. Ritschard, "An asymmetric entropy measure for decision trees", IPMU 06, Paris, France, July 2006.
13. B. Kavsek, N. Lavrac, L. Todorovski, "ROC Analysis of Example Weighting in Subgroup Discovery", European Conference on Artificial Intelligence, 2004.
14. S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, "Optimization by Simulated Annealing", Science Volume 220, Number 4598, 1983.
15. S. Hettich and S.D. Bay, "The UCI KDD Archive", Irvine, University of California, USA, Department of Information and Computer Science, 1999.

Improving SVM-Linear Predictions Using CART for Example Selection

João M. Moreira¹, Alípio M. Jorge², Carlos Soares³, and Jorge Freire de Sousa⁴

¹ Faculty of Engineering, University of Porto, Portugal
`jmoreira@fe.up.pt`

² Faculty of Economics, LIACC, University of Porto, Portugal
`amjorge@liacc.up.pt`

³ Faculty of Economics, LIACC, University of Porto, Portugal
`csoares@liacc.up.pt`

⁴ Faculty of Engineering, University of Porto, Portugal
`jfsousa@fe.up.pt`

Abstract. This paper describes the study on example selection in regression problems using μ -SVM (Support Vector Machine) linear as prediction algorithm. The motivation case is a study done on real data for a problem of bus trip time prediction. In this study we use three different training sets: all the examples, examples from past days similar to the day where prediction is needed, and examples selected by a CART regression tree. Then, we verify if the CART based example selection approach is appropriate on different regression data sets. The experimental results obtained are promising.

1 Introduction

The performance of prediction algorithms can be improved by selecting the examples used for training. The improvement of performance may correspond to faster training by reducing the number of examples and/or better predictions by selecting the most informative examples [1].

In this paper we approach the problem of example selection (or instance selection) mainly to improve predictive performance.

In section 2 we present results using three different training sets for μ -SVM with the linear kernel and with different parameter sets. All these experiments were done using bus trip time data. In section 3 we present the results of using CART's leaf node (one of the two example selection techniques described in section 2) on different regression data sets collected from L. Torgo's repository ([11]) and compare them with the ones from μ -SVM linear without example selection for different parameter sets. We also compare both algorithms using the corresponding best parameter set with CART and linear regression. In section 4 we discuss the results obtained and in the next section we review related work. We end with a review of the work done and a discussion of the guidelines for future work.

2 Example Selection for Bus Trip Time Prediction

In this section we present results of example selection applied to a real decision support problem of bus trip time prediction. This problem consists in predicting three days ahead the duration of a particular urban bus trip. The predicted trip times are then used to define crew duties and reduce costs in terms of extra time paid.

We present results using μ -Support Vector Machine ([8], [7]) with the linear kernel. Tests were done using data from January 1st 2004 to March 31st 2004, i. e., 2646 trip records. The training is done on a sliding window one month long and the test data is one day long. The lag between the train and test sets is three days long.

The explanatory variables used are: (1) start trip time (in seconds); (2) day type (bank holiday, normal, bridge or tolerance); (3) weekday; and (4) day of the year. The bridge day type corresponds to a day, usually a Monday or a Friday, respectively if a bank holiday falls on a Tuesday or a Thursday, that people take as holiday so people can enjoy a four-day weekend. The tolerance day type is similar to the bridge day type, but, in the tolerance case, this day was declared by the government a day that civil servants can take off. The start trip time and the day of the year are numeric while the other two are symbolic. The target variable is the trip time duration and is a numeric one. The series is irregularly time spaced with 29,5 trips per day in average. See [6] for a more complete description of the bus trip time data set.

On these trip time data, we have tried 3 different methods for example selection:

1. All: use all the examples from the same bus line;
2. Equivalent days (ed): use the examples from the same bus line and from identical past days;
3. Leaf node (ln): use the examples from the same bus line and from the same leaf node of a CART (CART - Classification And Regression Trees [2]).

Equivalent days were defined by visual inspection and using experts knowledge. For each bus line were defined nine groups of equivalent days:

- Day Type = Normal and working days (from Monday to Friday)
- Day Type = Normal and Saturdays
- Sundays
- Day Type = bank holiday and weekday on Monday, Friday
- Day Type = bank holiday and weekday on Tuesday, Thursday
- Day Type = bank holiday and Wednesday
- Day Type = bank holiday and Saturdays
- Day Type = bridge day
- Day Type = tolerance day

The example selection using the CART leaf nodes was done by first obtaining a CART regression tree over the initial training set. To predict the time for a

new trip, we identify the leaf node of the tree where the new trip would belong and obtain all the elements from this leaf node. The members of the training set are the members from the selected leaf node. We have used the rpart (from R-project [10]) with the default parameters. We use the mean squared error (mse) as evaluation criterion.

Results for different parameter sets are presented in fig. 1. The parameter sets were obtained by combining the two dimensional parameter space (C and μ) where C ranges from $2^{(2*(-2))}$ to $2^{(2*6)}$; and μ ranges from $1/10$ to $10/10$. The numbers in bold face will be referred as C index and μ index, respectively.

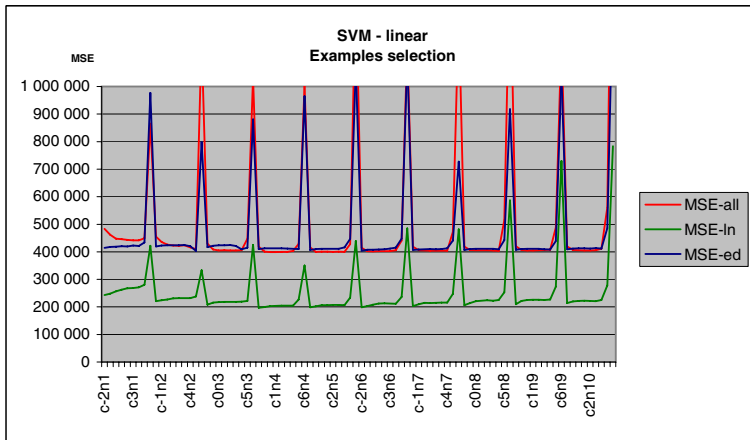


Fig. 1. Example selection for μ -SVM linear. The x-axis's values are expressed as $c[C \text{ index}]n[\mu \text{ index}]$.

From the analysis of the results we can see that the best results for all parameter sets are obtained using the leaf node method. These results were the motivation for the study that we describe in the next section.

3 Using CART's Leaf Node Members as Training Set for μ -SVM Linear on Regression Data Sets

In this section we validate if the positive results, obtained in the previous section for μ -SVM linear using the CART's leaf node approach for example selection, can be generalized to the regression problem. To run these tests we have collected eleven regression data sets from the L. Torgo's repository ([11]). We have selected data sets of diverse sizes, with the largest one, abalone, having more than 4000 examples.

The experimental procedure (figure 2, adapted from [9]) uses 10-fold cross validation. The selection task was performed by the CART's leaf node approach

described in the previous section. For each fold, the CART runs once. The parameters tuning was performed by varying C from $2^{(2*(-4))}$ to $2^{(2*4)}$; and μ from $(2 * 0 + 1)/10$ to $(2 * 4 + 1)/10$. The 45 parameter sets for both training sets (with and without selection) run over the same 10 folds previously generated, i. e., we are generating paired samples. Pre-all and Pre-ln are, each one, a set of mean squared error values, one for each parameter set tested.

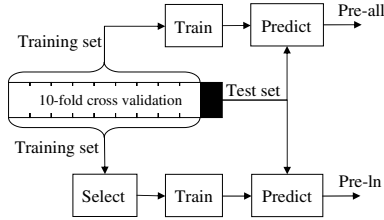


Fig. 2. Experimental procedure

This process was executed 5 times for each data set. In each of the 5 runs, different folds are obtained in 10-fold used for cross-validation. Figures from 3 to 8 present the results.

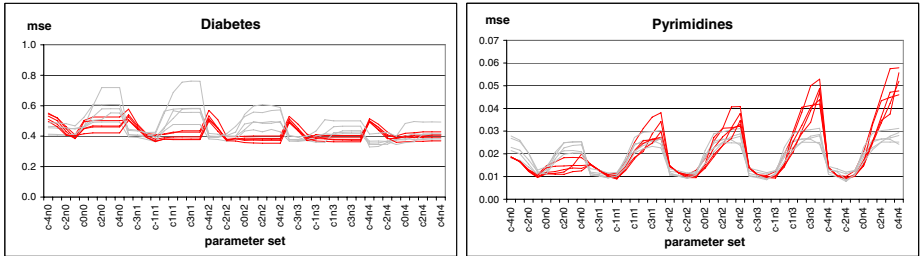


Fig. 3. Diabetes and pyrimidines data sets. The 5 dark lines represent the Pre-all results and the 5 clear lines represent the Pre-ln results. The x-axis's values are expressed as $c[C \text{ index}]n[\mu \text{ index}]$.

Figure 9 shows how much relative time is spent using the example selection technique compared with the values obtained using all data. The data sets are ordered by number of examples (ascending order). The values associated to Pre-all are calculated doing the ratio of the average time to run the 5 Pre-all results over the average time to obtain the 5 Pre-ln results.

Further tests were executed using the parameter set, from the 45 we have tested, with the best average for the 5 observations of Pre-all and the equivalent for Pre-ln. Results using the best parameter set from Pre-all and Pre-ln will be referred as all and ln respectively. We have also compared these results with

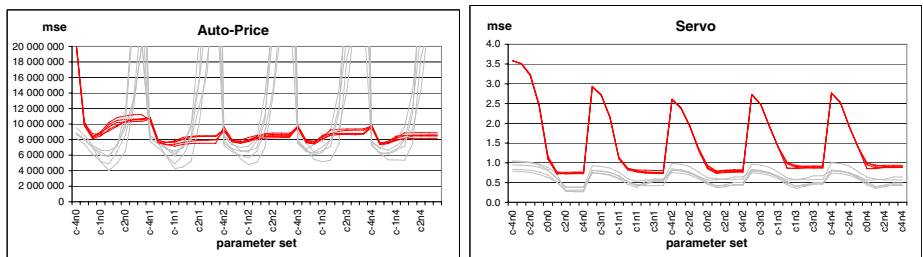


Fig. 4. Auto-price and servo data sets

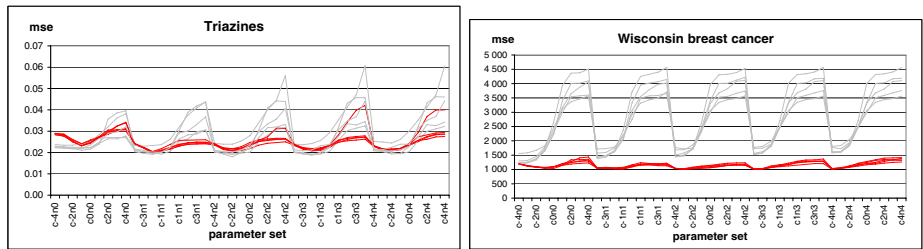


Fig. 5. Triazines and Wisconsin breast cancer datasets

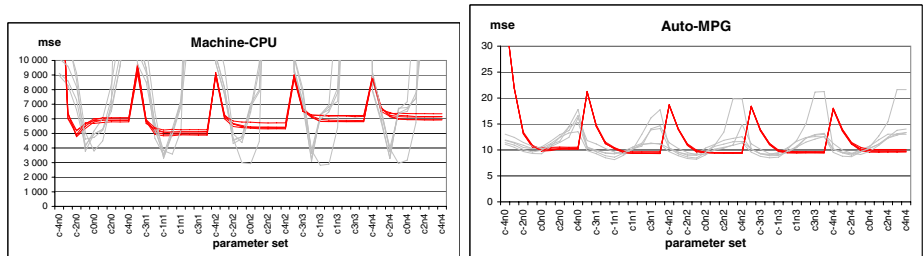


Fig. 6. Machine CPU and auto-MPG data sets

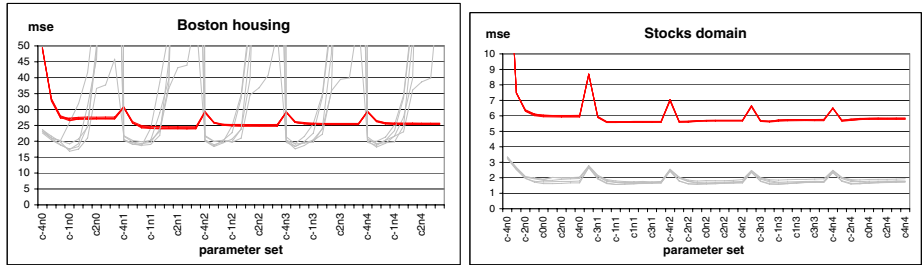


Fig. 7. Boston housing and stocks domain data sets

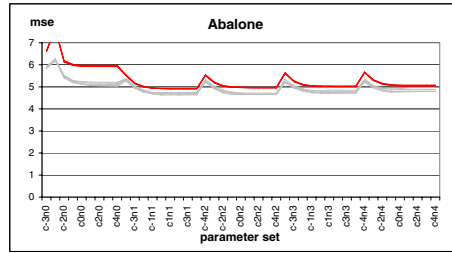


Fig. 8. Abalone data set

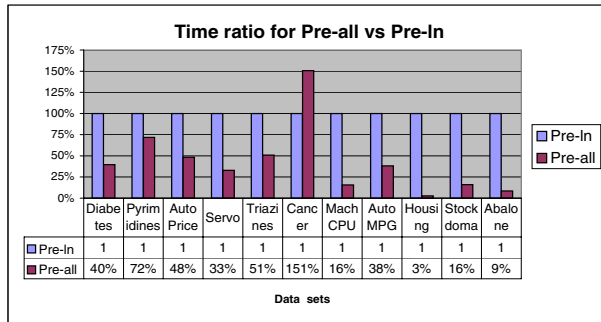


Fig. 9. Time ratio Pre-all vs Pre-ln

CART and linear regression. We got 50 observations for each technique using cross-validation but, this time, the samples are not paired. The lowest mse value for each data set is in bold face. We have used the default parameter set for rpart (CART) and lm (linear regression) functions from [10] (see figure 10). Lm results for the pyrimidines and triazines data sets are not reported because prediction from a rank-deficient fit may be misleading. Sd represents the standard deviation. The p-value for lm, CART and all refer to the p-value obtained on a hypothesis test about the difference between that technique mean and the mean of the ln results. The alternate hypothesis is to verify if the ln mean is lower than the mean of the method being compared. We use a type I error of 5%. The p-value is in bold face when is lower than 5%, i.e, when the null hypothesis is rejected.

Figure 11 compares execution times between the four tested methods.

4 Discussing Results

The first tests (presented in figures 3 to 9) intended to tune the parameters in order to select one parameter set for each data set. The tuning was done for two different approaches previously reported (Pre-all and Pre-ln). Some considerations must be done about these tests and their results.

- In general, results are more sensitive to C than to μ parameter values.
- The sensitivity to the parameter sets is very different between data sets and, for some data sets, is also very different between both tested techniques.
- There are several factors affecting time performance results (figure 9). In part, Pre-ln execution time is higher because of CART, even if its complexity is lower than in SVM’s case. The use of CART reduces the training examples, reducing the time to train SVM. However, the number of SVM’s trained has (if our algorithm was optimized, which is not the case), as upper bound, the number of the CART leaf nodes. In our case we train a new model when the value to predict falls in a different leaf node than the previous value to predict, increasing the number of trained models.
- The time to train a model depends strongly on the parameter set we are using. The higher the values of C and μ parameters, the slower is the training time. This is particularly evident in the C parameter case.

The analysis of results for Pre-all and Pre-ln was done using the Friedman rank sum test since the Kolmogorov-Smirnov Lilliefors test rejects the normality hypothesis and, consequently, it was not possible to use the ANOVA method with just 5 elements for each group. We used as blocks the eleven data sets

Data Set	lm			CART			all			ln	
	mean	sd	p-value	mean	sd	p-value	mean	sd	p-value	mean	sd
Diabetes	0.397	0.020	0.2%	0.393	0.035	2.1%	0.384	0.019	15.7%	0.377	0.044
Pyrimidines				0.0122	0.00094	0.0%	0.0093	0.00041	100.0%	0.0101	0.00158
Auto Price	7675516	363507	0.0%	8264533	735163	0.0%	7445161	169687	0.0%	5215136	787555
Servo	0.71	0.023	0.0%	0.77	0.073	0.0%	0.73	0.020	0.0%	0.39	0.120
Triazines				0.0217	0.00193	0.0%	0.0200	0.00032	21.7%	0.0198	0.00194
Cancer	1147	41.0	100.0%	1480	94.2	0.0%	1020	12.8	100.0%	1278	71.6
Mach CPU	4951	619	0.0%	10100	1110	0.0%	5001	279	0.0%	4076	1340
Auto MPG	9.2	0.159	6.2%	11.5	0.619	0.0%	9.4	0.159	0.0%	9.1	0.558
Housing	23.8	0.297	0.0%	23.3	1.240	0.0%	24.5	0.389	0.0%	18.8	2.183
Stock domain	5.51	0.023	0.0%	3.93	0.075	0.0%	5.60	0.029	0.0%	1.64	0.126
Abalone	4.91	0.014	0.0%	5.86	0.054	0.0%	4.91	0.011	0.0%	4.69	0.056

Fig. 10. Comparing different algorithms

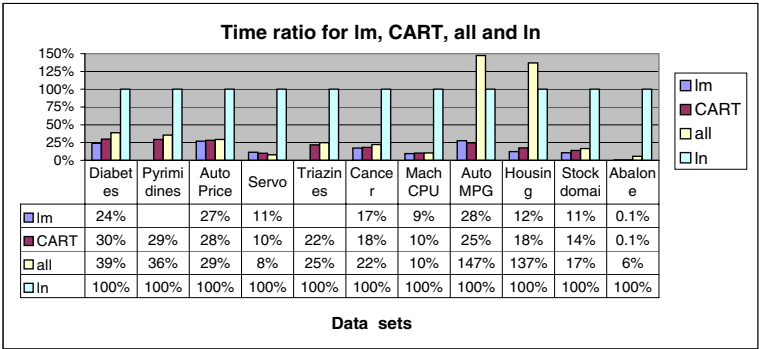


Fig. 11. Time ratio for lm, CART, all and ln

Table 1. Friedman rank sum test partial results

Alg	ps	Ri	Alg	ps	Ri	Alg	ps	Ri	Alg	ps	Ri	Alg	ps	Ri	Alg	ps	Ri
ln	c-1n1	173	all	c-1n2	357	all	c-1n3	420	all	c-3n2	518	all	c0n0	587	ln	c3n1	682
ln	c-2n3	176	all	c-2n2	368	ln	c-4n3	423	all	c0n4	519	all	c2n4	588	all	c-4n4	683
ln	c-2n4	195	ln	c-1n0	370	all	c-2n3	425	all	c-3n3	520	all	c-2n0	597	all	c-4n3	686
ln	c-2n2	197	all	c1n1	380	all	c3n2	431	all	c-1n0	523	all	c3n4	603	ln	c3n4	696
ln	c-1n2	210	all	c0n2	381	all	c4n2	443	ln	c1n0	523	ln	c-4n0	604	all	c-4n2	700
ln	c-2n1	229	ln	c-3n1	381	ln	c0n4	445	all	c2n3	531	all	c2n0	605	ln	c3n3	702
ln	c-1n3	247	all	c1n2	386	all	c-1n4	460	all	c3n3	536	all	c4n4	613	ln	c3n0	706
ln	c-1n4	268	ln	c0n0	392	ln	c-4n2	462	all	c4n3	549	ln	c2n1	625	ln	c4n1	706
ln	c-3n3	293	all	c2n1	395	all	c-2n4	465	all	c-3n1	550	ln	c2n2	625	ln	c4n2	709
all	c-1n1	297	ln	c0n2	401	all	c0n3	475	ln	c1n2	555	all	c3n0	627	ln	c4n4	721
ln	c-3n4	312	all	c3n1	402	ln	c-4n1	492	all	c1n4	566	all	c4n0	637	ln	c4n3	727
ln	c0n1	322	all	c4n1	407	ln	c-2n0	493	ln	c1n3	568	ln	c2n0	646	all	c-4n1	729
ln	c-3n2	330	ln	c0n3	413	all	c1n3	500	ln	c-3n0	569	ln	c2n3	646	ln	c4n0	740
all	c0n1	345	ln	c-4n4	416	ln	c1n1	508	all	c1n0	582	ln	c2n4	650	all	c-3n0	746
all	c-2n1	357	all	c2n2	417	all	c-3n4	516	ln	c1n4	584	ln	c3n2	674	all	c-4n0	817

tested, and as groups the combination of the algorithm and the parameter set used (2 algorithms $\times$ 45 parameter sets). As observation we used the average of the 5 existing observations in each group. The ranking table is presented in table 1 where Ri is the rank sum for the 11 data sets. The null hypothesis for this test is that the group effect is equal for all the groups. The null hypothesis is rejected with p value = 0.00%. This result allows us to say that there are meaningful statistical differences between the different algorithms / parameter sets.

We also observe, in Table 1, that the 9 top ranking models were obtained using the ln approach. It is also a tendency (although with exceptions) that for the same set of parameters, the model obtained with the ln example selection strategy ranks higher than the model obtained with the all strategy.

We have then compared results of four different algorithms: the two previously discussed with the best parameter set pre-selected for each data set, the CART and the linear regression. Some considerations must be done about these tests and their results.

- Time execution for ln and all (figure 11) are not equivalent to previous results (see figure 9) for all data sets as, for example, the Wisconsin breast cancer, auto-MPG and Boston housing data sets. This happens when the best parameter set for each one of the methods are very different, in particular for the C parameter. As we have previously said, the execution time increases strongly for higher parameter values. This is particularly true for the C parameter.
- With the exception of the pyrimidines and the Wisconsin breast cancer data sets, the best results using CART's leaf node members as training set are similar or better than without example selection.

- For the data sets auto-price, servo, machine-CPU, Boston housing, stocks domain and abalone results using example selection (ln) are meaningfully better compared to all other techniques. For the diabetes and triazines data sets the results are meaningfully better for ln comparing to all other techniques except for μ -SVM linear without example selection (p values of 15.7% and 21.7%, respectively). For the auto-MPG data set the ln method is not statistically better than the linear model (p value of 6.2%) but it is for the other two methods (all and CART). Ln performs badly for the Wisconsin breast cancer data set (looses for the all and the linear models) and also for the pyrimidines data set (looses for the all method).
- Ln performs always better than CART. This can be explained by the fact that, in spite of the similarity of the approaches, the prediction model is the average in CART's case and the μ -SVM linear in ln's case (as expected, μ -SVM linear performs better than the average for all data sets).
- Ln variance is much higher than all variance. This can be explained by CART induction algorithm (compares standard deviation values between ln and CART) and also by the reduction of training data in ln method.

5 Related Work

In this paper we present a study on example selection in order to improve predictive accuracy ([1]). Other authors use example selection in order to reduce training time ([5]). The works on example selection and feature selection have many aspects in common ([1]). Techniques for feature selection are usually classified as filters or wrappers ([4]). Filter techniques select the features independently of the used induction algorithm while the wrapper approaches have an embedded evaluation of the selection according to the used induction algorithm ([4]). Using this classification we can say that our approach is a filter one. A similar approach had been tried before in [9]. In that work, the motivation was to test if the use of the support vectors from SVM as training set is an example selection technique that is model independent, i. e., if it can improve results independently of the prediction algorithm chosen. The results obtained in [9], however, are not as expected, since none of the used algorithms (multi-layer perceptrons, nearest neighbors and the C4.5 algorithm for decision trees) improves predictions. Moreover, different algorithms obtain different results. We believe that filter techniques are not model independent, i.e., there is no guarantee that the best training set for a certain induction algorithm is the best one for another induction algorithm. However, we believe that a wrapper approach may be model independent. The wrapper approach has the additional advantage of giving better results than the filter approach. It has the disadvantage of being computationally heavier. Both filter and wrapper approaches are worth researching in the future.

The use of decision trees for feature selection is known ([3]), but we do not know any work using decision trees for example selection.

6 Conclusions and Future Work

This paper describes a study on example selection for μ -SVM linear. Firstly we describe two approaches for example selection on a bus trip time prediction task. We have done tests using μ -SVM with the linear kernel. We observed the impact of example selection for this data set. Then, we tried using as training examples for μ -SVM linear the ones on the same CART's leaf node as the case being predicted. This technique was then tested on other 11 regression data sets. Results show that, with the exception of two data sets, all the others give equal or lower mean squared error when using the example selection technique. This is a very promising result.

Future work will include the reduction of execution time by optimizing our algorithm and by testing faster decision trees induction algorithms, the repetition of these tests for other SVM kernels (we also got promising results using μ -SVM with the gaussian and the sigmoidal kernels for the bus trip time data set), and the tuning of the decision tree induction algorithm parameters.

Acknowledgments

This work was partially supported by FCT - Fundação para a Ciência e a Tecnologia, project reference: POCT/TRA/61001/2004 and FEDER e Programa de Financiamento Plurianual de Unidades de I&D.

References

1. A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.
2. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Chapman and Hall/CRC, 1984.
3. C. Cardie. Using decision trees to improve case-based learning. In *10th International conference on machine learning*, pages 25–32. Morgan Kaufmann, 1993.
4. R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.
5. H. Liu and H. Motoda. On issues of instance selection. *Data Mining and Knowledge Discovery*, 6(2):115–130, 2002.
6. J. M. Moreira, A. Jorge, J. F. Sousa, and C. Soares. Trip time prediction in mass transit companies. a machine learning approach. In *10th EWGT*, pages 276–283, 2005.
7. B. Scholkopf, A. J. Smola, R. Williamson, and P. Bartlett. New support vector algorithms. Technical Report NC2-TR-1998-031, 1998.
8. A. J. Smola and B. Scholkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, 1998.
9. N. A. Syed, H. Liu, and K. K. Sung. A study of support vectors on model independent example selection. In *5th ACM SIGKDD*, pages 272–276, 1999.
10. R. D. C. Team. R: A language and environment for statistical computing. Technical report, R Foundation for Statistical Computing, 2004.
11. L. Torgo. Regression data repository, <http://www.liacc.up.pt/~ltorgo>.

Simulated Annealing Algorithm with Biased Neighborhood Distribution for Training Profile Models

Anton Bezuglov¹ and Juan E. Vargas²

¹ University of South Carolina, Columbia SC 29205
anton@esri.sc.edu

² Microsoft Co., Redmond WA 98052
juanv@microsoft.com

Abstract. Functional biological sequences, which typically come in families, have retained some level of similarity and function during evolution. Finding consensus regions, alignment of sequences, and identifying the relationship between a sequence and a family allow inferences about the function of the sequences. Profile hidden Markov models (HMMs) are generally used to identify those relationships. A profile HMM can be trained on unaligned members of the family using conventional algorithms such as Baum-Welch, Viterbi, and their modifications. The overall quality of the alignment depends on the quality of the trained model. Unfortunately, the conventional training algorithms converge to suboptimal models most of the time. This work proposes a training algorithm that early identifies many imperfect models. The method is based on the Simulated Annealing approach widely used in discrete optimization problems. The training algorithm is implemented as a component in HMMER. The performance of the algorithm is discussed on protein sequence data.

1 Background

Hidden Markov models (HMMs) have proven their value in different applications in Molecular Biology [1, 2]. Durbin et al., Goutsias, and Krogh et al. propose in [3, 4, 5] the use of HMMs as formal mathematical models of *profiles*. The profile represents a *structure consensus* [6, 7] of proteins that share identical functional role and belong to one single family. Profile HMMs are Markov models with a topology that represents inserted, deleted, and matching regions in proteins. Hidden Markov models provide a rich, conceptually full, probabilistic framework for expressing profiles.

A profile HMM can be derived from a known multiple alignment of proteins, where all homologous positions match. However, finding the multiple alignments may be a labor-intensive work in most real-world biological sequences [1]. An appealing property of profile HMMs is that it is possible to train them from protein sequences. The profile HMMs can be *trained* from unaligned proteins, when the multiple alignment is not known [3, 4].

The classical training algorithm for ordinary HMMs is the Baum-Welch algorithm [8,9,10,11], which is a form of the Expectation-Maximization (EM) technique [12,8]. Other options for training include the Viterbi algorithm [13,14], the gradient algorithm [15], or any optimization method for a nonlinear goal function.

The common limitation of the traditional training algorithms is their convergence to suboptimal models. When a suboptimal profile HMM is found, it means that the model may not have captured all the features of the protein consensus [1].

Different techniques are applied to avoid suboptimal models during the training. A physical phenomenon of crystallization is used as an insight for Simulated Annealing (SA) [16,17,18]. It is known that some compounds crystallize to *minimum energy* structures, if the temperature is cooled slowly. SA is a mathematical approximation of crystallization, where an artificial temperature parameter is added to the optimized system. The goal function, which is maximized or minimized, is expressed as the internal energy.

The application of SA in training HMMs is not new. An approach to model SA by using *noise injection* has been proposed in [3,19]. In this method, HMM parameters are modified by noise, the intensity of which depends on a temperature parameter. Another approach of discrete SA application is the *simulated annealing Viterbi estimation* [5]. In this method, noised model parameters are used in the Viterbi estimation algorithm. As a result, a suboptimal trajectory path is calculated on every iteration and the model parameters are adjusted accordingly.

In contrast to the classical procedure of SA [16], neither the noise-injection, nor SA Viterbi estimation use *acceptance probability* [16]. The acceptance probability appears to be a quite important step of SA, since it makes possible formal theoretical proofs of the algorithm's asymptotic convergence [18].

In contrast to the previous SA-inspired approaches, this paper proposes a training algorithm that implements SA for training Profile HMMs in a more formal way. Conditions that guarantee the global convergence of the algorithm at finite time can be found. The proposed algorithm, SA with Biased Neighborhood Distribution (SA-BN) algorithm uses the discrete SA for Viterbi Estimation. The algorithm is the result of applying techniques from [5] in [18,16].

2 The Biased Neighborhood Distribution Simulated Annealing (BN-SA) Algorithm

2.1 Profile Hidden Markov Models

Profile Hidden Markov models have a different topology than ordinary HMMs. A typical profile HMM [3] with 3 consensus positions, is shown in Figure 1. The hidden states in profile models are divided into 3 groups: match states 'M'; insert states 'I'; and delete states 'D'. A profile HMM is defined by two probability distributions: *emission probability* and *transition probability*.

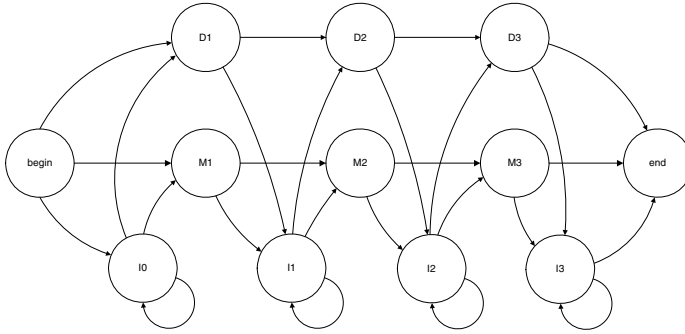


Fig. 1. A Profile Hidden Markov Model for 3 symbol consensus

The profile model activates at the ‘begin’ state. Depending on the amino acid at the first position of a protein sequence, the model proceeds to ‘M1’, ‘I0’, or ‘D1’. A transition to ‘M1’ is made, if the amino acid matches the first symbol in the profile. Otherwise, the transition is made to ‘I0’ or ‘D1’. When ‘delete’ states are visited, the current position in the sequence remains unchanged. The analysis is finished when the model is in the ‘end’ state.

The explicit designation of the hidden states to matches, inserts and deletes makes ordinary the multiple alignment of proteins. The multiple alignment of a protein O to the profile is the sequence of state assignments or the state trajectory. The state trajectory can be found by using Viterbi estimation procedure [8].

The score of the alignment is represented by the *likelihood* (or logarithm of the likelihood) of O , which can efficiently be calculated by the *forward procedure* [8]. The score gives an estimate of how well the protein fits in the family. The summarized score over all members indicates the quality of the modeled consensus and is used as the *goal function* in the training.

The training of the profile HMM is the search for a model that maximizes the alignment scores of all protein sequences from the family. This optimization is performed in the space of possible alignments, which is discrete and finite, so the discrete version of SA can be applied.

2.2 The Simulated Annealing Algorithm

The Simulated Annealing (SA) Algorithm makes possible finding global extremum of some nonlinear goal function $f(x)$, where x represents the vector of parameters. The *maximum likelihood (ML)* criterion is usually chosen as the goal function for HMM training, the SA algorithm is applied to find the ML of the profile model.

The typical form of the SA algorithm [18] is given in Figure 2. On each iteration, the algorithm finds new states of the model $x \rightarrow x'$, which are accepted or rejected with *acceptance probability* $A_T(x, x')$. A new iteration starts when state x' is accepted. Probably, the most important components are the method of generating new states and the acceptance probability rule.

1. Set starting state $x = x_0$
2. Set temperature $T = T^0$ and cooling rate $0 < \alpha < 1$
3. while stopping condition not satisfied
4. for $k = 1$ to N_t do
5. generate trial point x' from S_x using $q(x, x')$
6. accept x' with probability $A_T(x, x')$
7. end for
8. reduce temperature $T = T \cdot \alpha$
9. end while

Fig. 2. Simulated Annealing (SA) algorithm

Types of Acceptance Probabilities. The acceptance probability $A_T(x, x')$ controls how new states x' are accepted or rejected. It is done by favoring (higher probabilities) some states and ignoring (lower probabilities) others. There is a variety of recommended acceptance probabilities: Metropolis rule (1), Logistics rule [20, 21], Hastings' rule [22, 23] and Tsallis' rule [24, 25].

$$A_T(x, x') = e^{-\frac{(f(x') - f(x))^+}{T}} \quad (1)$$

The key feature of the acceptance rule can be illustrated by Metropolis rule (1), where $(a)^+ = a$, $a > 0$; $(a)^+ = 0$, $a \leq 0$. The rule always supports transitions ($A_T(x, x') = 1$) to states x' , where the goal function is better $f(x') \leq f(x)$. Transitions to states with lower goal function values still have positive acceptance probability $0 < A_T(x, x') < 1$. The greater the temperature T and less the $f(x) - f(x')$, the higher the probability is. Unless the temperature equals to 0, there is always positive probability of leaving a local optimum. This makes the SA algorithm different to various hill-climbing and gradient-type algorithms that cannot leave local optima.

In our training algorithm we use the basic Metropolis rule. However, other acceptance probability rules may add performance to the algorithm and further research may result in better acceptance strategies, specific to HMM training.

Types of Neighborhood Distributions. New states in SA are sampled from the neighborhood distribution $q(x, x')$. It is critical for $q(x, x')$ to make all the states *reachable* at finite time. Otherwise, the global optimum may be missed. The following neighborhood distributions are generally applied: uniform [26], normal [27, 28] and Cauchy [28] distributions.

The listed distributions have parameters that may be controlled by SA during its work. For instance, a 1 : 1 rule [26] adjusts the standard deviation σ . According to the rule, when too many configurations are rejected, σ is decreased. This results in the search among states that are similar to x . If trial points are accepted too often, then σ is incremented and the new state x' will be different from x . This feedback mechanism diminishes the number of iterations needed for the convergence, since totally random choice walks on the parameter space may not be feasible.

However, even with optimization techniques like 1 : 1 rule, the convergence to the optimum may take an intractable number of iterations, because the standard distributions provide a purely random walk in the parameter space. We propose to sample new states from the distribution of possible state trajectories in the profile HMM $P(x|O, \theta)$ (2).

$$P(x|O, \theta) = \frac{P(O, x|\theta)}{P(O|\theta)} \quad (2)$$

$P(x|O, \theta)$ gives the distribution of all the possible alignments of a sequence O . Given the model θ , some alignments are better than others and thus have higher probabilities. As a result, sampling from the distribution favors state trajectories x , which represent better alignments of O . Worse alignments are visited rarely, but are still reachable.

In order to control the shape of distribution (2), the HMM parameters θ are noised. This approach has been inspired from HMMER [5]. An analog of temperature τ modifies the parameters of HMM so that high values of τ make (2) almost uniform. So, τ is analogous to σ from the neighborhood distributions.

The direct application of (2) is not practical because of the size of X . A computationally effective method of sampling from the trajectory distribution is the application of the forward algorithm [8]. The sampling from the distribution is made as a sequence of T samples from distributions $P(x_{t-1}|x_t)$. Sequentially sampling from distributions $P(x_{t-1}|x_t)$, the sampling from (2) is obtained.

2.3 BN-SA Outline

The BN-SA algorithm is based on the general SA procedure Figure 2, acceptance probability (1) and the neighborhood distribution (2). The outline of the algorithm is given in Figure 3.

On each iteration k , the possible alignments x'_j for each of the protein sequences O_j are sampled and the maximum likelihood HMM is built θ' . The parameter τ is adjusted according to the acceptance ratio.

The implementation of the algorithm is done at the basis of HMMER package [5]. The BN-SA training is added as an option to other training algorithms.

3 Experiments and Discussion

To illustrate the performance of the BN-SA algorithm, a well studied family of *globin* proteins was chosen [29]. The family contains 1817 true positive protein sequences. The true alignment of the sequences is available for download from SwissProt sequence database.

We compared the BN-SA algorithm against SA-HMMER, the Baum-Welch and the Viterbi training algorithms. All the algorithms were run at their default settings. The SA-HMMER initial temperature $k = 5$ and ramp $r = 0.95$ were used. The identical cooling schedule was used for the BN-SA.

1. Set starting configuration $x = x_0$
2. Set temperature $T = T^0$ and cooling rate $0 < \alpha < 1$
3. while stopping condition not satisfied
4. for $k = 1$ to N_t do
5. $NotAcceptedCount = 0$
6. for $j = 1$ to N_O do
7. Repeat
8. generate x'_j for O_j from $q_\tau(x_j, x'_j)$
9. $NotAcceptedCount++$
10. Until point accepted $A_T(x_j, x'_j)$
11. end for
12. Given N_O of x'_j , find ML parameters θ'
13. Adjust τ , based on $NotAcceptedCount$
14. end
15. end for
16. reduce temperature $T = T \cdot \alpha$
17. end while

Fig. 3. BN-SA algorithm

3.1 Comparing Performance of Training Algorithms

To compare the performance of SA-HMMER, Baum-Welch, Viterbi, and BN-SA algorithm *alignment scores* are used. The alignment score represents the log likelihood of all proteins from the training set. The quality of the multiple alignment is quantitatively estimated by the score. The local convergence of the training algorithms causes the scores to vary from training to training. Hence we used a series of training experiments on the same dataset to find the *average* quality of the obtained alignments. Figure 4 shows the results of this experiment.

The training from Figure 4 was performed on a subset of 50 globin proteins. A series of 100 trainings were run and model scores were recorded. The average quality of models obtained by BN-SA is higher than that of Baum-Welch algorithm by 10% and SA-HMMER by 3%. Besides, the BN-SA algorithm demonstrates lower variability of the alignment scores, and lower risk of the local convergence.

3.2 Comparing Multiple Alignment

Here we illustrate that the models with greater scores provide better multiple alignments. Figure 5 shows 3 fragments of alignments. The alignment *a*) is the true alignment, downloaded from the protein sequence database. Alignments *b*) and *c*) were produced by profile models trained by SA-HMMER and BN-SA algorithms respectively. The score of the BN-SA model is higher than the score of the SA-HMMER model and (at least at this fragment) the multiple alignment is closer to the true alignment.

Another way of comparing alignments is by using the pair-wise distances between aligned sequences. Table 1 gives the pair-wise distances of a fragment of the alignment. Similar to the previous example, the score of the BN-SA model

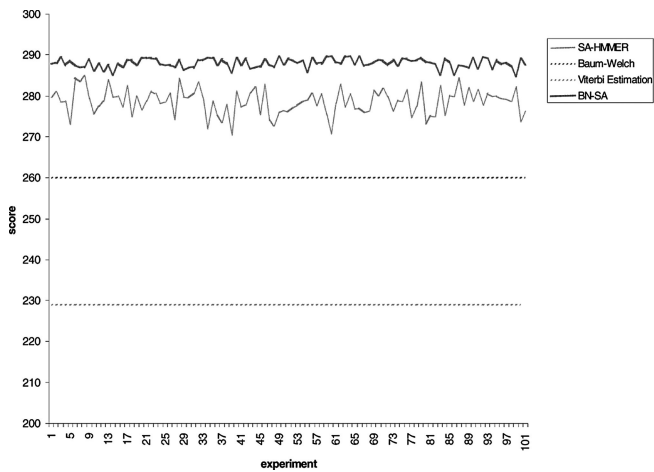


Fig. 4. Profile models, trained on 50 unaligned protein sequences from globins family

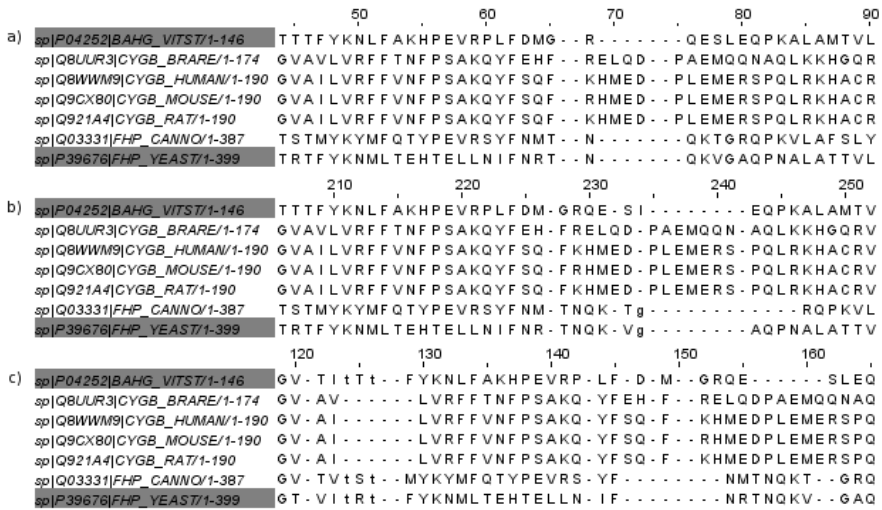


Fig. 5. Fragment of the globins alignment. a - true alignment, b - BN-SA, c - SA-HMMER.

is higher than the score of the SA-HMMER model. It can be seen from the table, that the BN-SA alignment gives more precise estimations of the pairwise distances.

Finally, Figure 6 demonstrate distance trees created from different multiple alignments. The figure shows that in case of those proteins, better estimation of pairwise distances results in more precise distance tree.

Table 1. Distances between aligned sequences. Top – true distances, middle – BN-SA, bottom – SA-HMMER.

	BAHG_VIT	CYGB_BRA	CYGB_HUM	CYGB_MOU
BAHG_VIT	0.000	0.873	0.881	0.865
CYGB_BRA	0.873	0.000	0.457	0.457
CYGB_HUM	0.881	0.457	0.000	0.050
CYGB_MOU	0.865	0.457	0.050	0.000

	BAHG_VIT	CYGB_BRA	CYGB_HUM	CYGB_MOU
BAHG_VIT	0.000	0.838	0.837	0.816
CYGB_BRA	0.838	0.000	0.526	0.515
CYGB_HUM	0.837	0.526	0.000	0.047
CYGB_MOU	0.816	0.515	0.047	0.000

	BAHG_VIT	CYGB_BRA	CYGB_HUM	CYGB_MOU
BAHG_VIT	0.000	0.809	0.794	0.787
CYGB_BRA	0.809	0.000	0.529	0.524
CYGB_HUM	0.794	0.529	0.000	0.047
CYGB_MOU	0.787	0.524	0.047	0.000

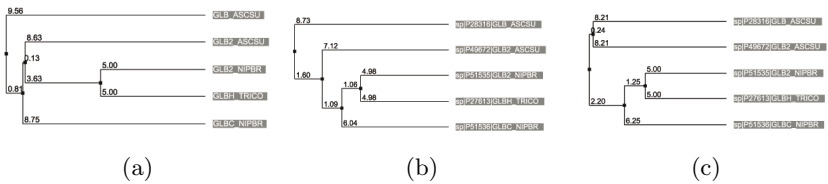


Fig. 6. Distance trees: (a) – true alignment; (b) – BN-SA; (c) – SA-HMMER

4 Conclusions

Profile HMMs are a formal probabilistic way of describing profiles. The automated creation of profile HMMs allows producing multiple alignments of protein sequences. However, this automated creation or training of profile models is a nonlinear optimization problem, in which global optima are rarely obtained. The training algorithms related in the literature are prone to local convergence that results in suboptimal profile models. Although our BN-SA algorithm in its current configuration does not guarantee global convergence, it is able to find profile models of better quality. Due to the usage of acceptance probability and biased neighborhood distribution, more locally optimal profile HMMs are ignored, which leads to better overall scores of alignments. The multiple alignments obtained are more accurate, and pair-wise distance between sequences are more precise, which in turn makes automated creation of profiles more effective.

Acknowledgements

We thank Sean Eddy and his team for the high-quality HMMER source code that we used for implementing the BN-SA algorithm.

References

1. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: *Biological Sequence Analysis*. Cambridge University Press (1998)
2. Goutsias, J.: A hidden Markov model for transcriptional regulation in single cells. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **3**(1) (2006) 57–71
3. Krogh, A., Brown, M., Mianx, S., Sjolander, K., Haussler, D.: Hidden Markov models in computational biology: Applications to protein modeling ucsc-crl-93-32. *Journal of Molecular Biology* (235) (1994) 1501–1531
4. Baldi, P., Chauvin, Y., Hunkapiller, T., McClure, M.A.: Hidden Markov models of biological primary sequence information. *Proc. Natl. Acad. Sci. USA* (91) (1994) 1059–1063
5. Eddy, S.R.: Multiple alignment using hidden Markov models. (In: *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*. AAAI Press.) 114–120
6. Barton, G.J.: Protein multiple sequence alignment and flexible pattern matching. *Meth. Enzymol.* (183) (1990) 403–427
7. Bashford, D., Chotia, C., Lesk, A.M.: Determinants of a protein fold: Unique features of the globin amino acid sequences. *Journal of Molecular Biology* **196**(1) (1987) 199–216
8. Rabiner, L.R.: Tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* **77**(2) (1989) 257–286
9. Baum, L.E., Petrie, T.: Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics* **37** (1966) 1554–1563
10. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics* **41**(1) (1970) 164–171
11. Baum, L.E.: An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities* (3) (1972) 1–8
12. Dempster, A.P., Laird, N.M., Rubin, D.: Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. Ser. B.* (39) (1977)
13. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory* **IT-13** (1967) 260–269
14. Forney, G.D.: The Viterbi algorithm. *Proceedings IEEE* **61** (1973) 268–278
15. Levinson, S.E., Rabiner, L.R., Sondhi, M.M.: An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *Bell Syst. Tech. J.* **62**(4) (1983) 1035–1074
16. Kirkpatrick, S., C. D. Gelatt, J., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598) (1983) 671–680
17. Ingber, L.: *Adaptive Simulated Annealing (ASA)*. Lester Ingber Research (1995)
18. Wang, T.: *Global Optimization for Constrained Nonlinear Programming*. PhD thesis, University of Illinois (2001)
19. Hughey, R., Krogh, A.: Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *Computer Applications in the Biosciences* (12) (1996) 95–107
20. Aarts, E., Korst, J.: *Simulated Annealing and Boltzmann Machines*. J. Wiley and Sons (1989)

21. Romeo, F., Sangiovanni-Vincentelli, L.: A theoretical framework for simulated annealing. *Algorithmica* (6) (1991) 302–345
22. Hastings, W.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* (57) (1970) 97–109
23. Fogel, D.B.: An introduction to simulated annealing evolutionary optimization. *IEEE Trans. on Neural Networks* **5**(1) (1994) 3–14
24. Andricioaei, I., Straub, J.E.: Generalized simulated annealing algorithms using Tsallis statistics: Application to conformational optimization of a tetrapeptide. *Physical Review E* **53**(4) (1996) R3055–R3058
25. Hansmann, U.H.E.: Simulated annealing with Tsallis weights: A numerical comparison. *Physical A* (242) (1997) 250–257
26. Corana, A., Marchesi, M., Martini, C., Ridella, S.: Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. *ACM Trans. on Mathematical Software* **13**(3) (1987) 262–280
27. Courrieu, P.: The hyperbell algorithm for global optimization: A random walk using Cauchy densities. *Journal of Global Optimization* (10) (1997) 37–55
28. Yao, X., Lin, G.: Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation* **3**(2) (1999) 82–102
29. Scott, T.A., Mercer, E.I.: *Concise Encyclopedia Biochemistry and Molecular Biology*, Third Edition. Walter de Gruyter, Berlin New-York (1998)

A Conditional Model for Tonal Analysis

Daniele P. Radicioni and Roberto Esposito

Università di Torino, Dipartimento di Informatica
C.so Svizzera 185, 10149, Turin, Italy
{radicion, esposito}@di.unito.it

Abstract. Tonal harmony analysis is arguably one of the most sophisticated tasks that musicians deal with. It combines general knowledge with contextual cues, being ingrained with both faceted and evolving objects, such as musical language, execution style, or even taste. In the present work we introduce BREVE, a system for tonal analysis. BREVE automatically learns to analyse music using the recently developed framework of conditional models. The system is presented and assessed on a corpus of Western classical pieces from the 18th to the late 19th Centuries repertoire. The results are discussed and interesting issues in modeling this problem are drawn.

1 Introduction

Music analysis is arguably one of the most sophisticated tasks that musicians deal with. It combines general knowledge with contextual cues, being ingrained with both faceted and evolving objects, such as musical language, execution style, and even taste. Besides, analysis is a crucial step in order to deal with music. Performers, listeners, musicologists, automatic systems for performance, composition and accompaniment: all of them need to process and “understand” the pieces being composed, listened or performed. Consequently, analysis has been addressed by didactic literature [1]; psychological concerns have inspired theories about listeners perception [2], as well as about performers strategies and constraints [3]. Such seminal studies have been exploited to the end of devising automatic expressive performers in the AI field (see, e.g., [4,5]).

Within the broader music analysis area, we single out the task of tonal harmony analysis. Analyzing music harmony consists of associating a label to each *vertical* (that is, set of simultaneous notes) [6,7]. Such labels explain which harmony is sounding by indicating a chord name in terms of a *root* (the fundamental note) and a *mode*, such as C minor (Fig. 1).

Tonal harmony theory encodes how to build chords (i.e., which pitches compose each chord) and how to concatenate them. This task is, in general, a difficult one: in fact, music students spend considerable amounts of time in learning tonal harmony, which is still the base of both classical composer’s and performer’s background. Moreover, we have to handle ambiguous cases [8], and composer’s strategies aiming at surprising those of the listener, by suddenly introducing elements that violate expectation. In fact, to some extent musical language has



Fig. 1. The tonal harmony analysis problem consists of indicating for each vertical which chord is currently sounding. Excerpt from Beethoven’s Piano Sonata Op.10 n.1.

evolved under such a pressure for breaking “grammatical” rules [9]. This heightens the actual complexity of analysis, and also the difficulties encountered by automatic analysis systems.

It is worth mentioning two subtle points. Firstly, music *analysis* and *composition* tasks are conceptually different, though related. Analyzing music is introductory to composing music, and it plays a central role in systems for composition (see, e.g., the pioneering system by Ebcioglu [10]). In the present work, we do not describe the next artificial composer, but rather an analysis system. Secondly, an higher level of analysis (*functional* analysis) exists that aims at individuating harmonic functions in chords [9]. As it will be clearer later on, this kind of analysis is a long term goal for harmony analysis systems. In the present work, we address the basic form of harmonic analysis mentioned earlier, and similar to [6] and [7].

In summary, music analysis is a complex problem, requiring many hours of apprenticeship to professional musicians. This makes such knowledge difficult to be elicited, and enhances the difficulties encountered while devising systems to accomplish automatically the task. The present system attempts to tackle the problem with machine learning techniques, which also benefit from an encoding of harmony theory knowledge.

2 Related Works on Tonal Harmony Analysis

Much work has been carried out in the field of automatic tonal analysis: since the pioneering grammar-based work by Winograd [11], a number of approaches have been proposed that address the issue of tonal analysis.

One of the preference rules systems described by Temperley [6] is devoted to harmonic structure analysis. It relies on the Generative Theory of Tonal Music [2], providing that theory with a working implementation. In this approach, preference rules are used to evaluate possible kinds of analysis, such as harmony analysis, also exploiting meter, grouping, and pitch spelling information. A major feature of Temperley’s work is in applying high level domain knowledge,

such as “Prefer roots that are close to the roots of nearby segments on the line of fifths”, which leads to an explanation of results.

The system by Pardo & Birmingham [7] is a template matching system. It performs tonal analysis by assigning a score to a set of 72 templates (from the combination of 6 original templates transposed over 12 semi-tones of the chromatic scale). The resulting analysis is further improved by 3 tie-resolution rules, for labeling still ambiguous cases.

Raphael & Stoddard [12] propose a machine learning approach based on a Hidden Markov Model that computes roman numeral analysis (that is, the functional analysis mentioned above). A main feature of their system is that the generative model can be trained using unlabeled data, thus determining its applicability also to unsupervised problems. In order to reduce the huge number of parameters to be estimated, they make a number of assumptions, such as that the current chord does not affect the *key transitions*. Also, the generative model assumes conditional independence of notes (the observable variables) given the current mode/chord.

Our approach relies on the recent paradigm of *conditional models* [13]. Conditional models directly solve the conditional problem of predicting the label given the observations, instead of modeling the joint probability of observations and labels. We argue that—in the specific case of music *analysis*—the generative features of HMMs are neither required nor useful, as this actually results in forcing a more complex model to be estimated, without any additional benefit with respect to conditional models.

Unfortunately, experimental comparisons between harmony analysis systems are not that simple. The system by Temperley does not consider the mode of the chord (thus resulting in a different kind of analysis), while the algorithm used by Pardo is deeply affected by the accuracy of the segmentation. On the other hand, based on methodological accounts, our system should be compared with that from Raphael & Stoddard, although they do not provide results of a systematic experimentation.

3 The System

The chief idea behind our work is that music analysis task can be naturally cast to a Machine Learning (ML) problem, suitable to be solved by Supervised Sequential Learning (SSL) techniques. In fact, by considering only the “vertical” aspects of musical structure, one would hardly produce reasonable analyses. As mentioned in [14], experimental evidences about human cognition reveal that composers and listeners refer to “horizontal” features of music to disambiguate unclear cases. Therefore context plays a fundamental role and contextual cues are deemed useful to the analysis system. Moreover, harmony changes are well known to follow patterns implying that analysis must take into consideration the succession of chords (e.g., the case of *cadence*).

The fundamental representational structure processed by BREVE is the music *event*; whole pieces are represented as *event lists*. An event is a set of *pitch*

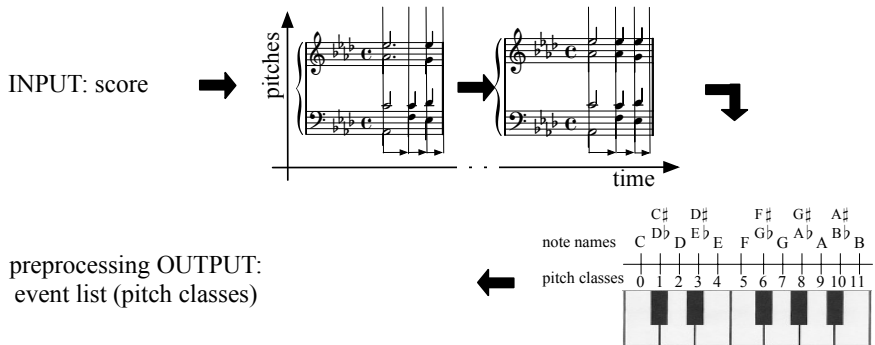


Fig. 2. Top: passing from the “Common Practice Notation”, displayed on the left side, to the corresponding “explicit” representation, on the right side. Bottom: notes are then converted into the corresponding pitch classes.

classes sounding at the same time. Any note onset or offset determines a new event. Pitch classes are categories “such that any two pitches one or more octaves apart are members of the same category” [6]. In other words, pitch classes are congruence classes under the modulo-12 relation. For example, a vertical composed by the notes C4-E4-G4, corresponding to the MIDI pitch numbers 60-64-67, is converted into an event composed by the pitch classes 0-4-7. We retain information about *duration* of events as well. For example, if a note i is held while a new note j is played, we consider an event containing i , and an event composed by both i and j since the held note affects the harmonic content of the new vertical as well (top of Figure 2).

Much of the information required to perform tonal harmony analysis lies in the notes currently sounding and in their metrical salience: specifically, accented events are expected to convey much harmonic content [2]. Hence for each event we not only encode which pitch classes are present, but also whether the event is accented or not. Meter estimation is based on the work of Temperley [6].

3.1 Learning Algorithm

To analyze the harmonic structure of a piece is a sequential task, where contextual cues are commonly acknowledged to play a fundamental role [2]. It follows that standard classification approaches such as decision trees, naive bayes, etc. are arguably not likely to produce good classification hypotheses, since they do not take into account the contextual information. This information is provided by surrounding observations, as well as nearby labeling.

The learning task consists in seeking an hypothesis H which accurately predicts the labels associated to events. The task, known as the Supervised Sequential Learning (SSL) task, is not novel to the Machine Learning community. A wealth of research has been invested to develop algorithms for solving this kind

of problems, and a number of interesting algorithms have been proposed [15,13]. The SSL task can be specified as follows [16]:

Given: A set L of training examples of the form (X_m, Y_m) , where each $X_m = (x_{m,1}, \dots, x_{m,T_m})$ is a sequence of T_m feature vectors and each $Y_m = (y_{m,1}, \dots, y_{m,T_m})$ is a corresponding sequence of class labels, $y \in \{1, \dots, K\}$.

Find: A classifier H that, given a new sequence X of feature vectors, predicts the corresponding sequence of class labels $Y = H(X)$ accurately.

In our case, each X_m corresponds to a particular piece of music; $x_{m,t}$ is the information associated to the event at time t ; and $y_{m,t}$ corresponds to the chord label (i.e., the chord root and mode) associated to the event sounding at time t . The problem is, thus, to learn how to predict accurately the chord labels given the musical events information. In case the specification of the identity of an example is not relevant, we drop the m subscript in the X_m notation. In such situations, an event at time t within a sequence X is denoted as $x_{\cdot,t}$.

The SSL problem can be solved with several techniques, such as Sliding Windows, Hidden Markov Models, Maximum Entropy Markov Models [17], Conditional Random Fields [18], and Collin's adaptation of the Perceptron algorithm to sequential problems [13] (henceforth, HMPerceptron). All these methods, with the exception of Sliding Windows, are generalizations and improvements of Markovian models, culminating in Conditional Random Fields and HMPerceptrons. Conditional Random Fields (CRFs) are state of the art conditional probabilistic sequence models, that improve on Maximum Entropy models [18,15] while maintaining most of the good properties of conditional models.

Unfortunately, the parameter estimation algorithm given for CRFs is known to be slow, in particular when a large number of features are to be handled [19]. A faster and simpler approach in learning conditional models has been recently proposed by Collins [13]. The algorithm, which is an extension to sequential problems of Rosenblatt's Perceptron algorithm, is reportedly at least on par with Maximum Entropy and CRFs models from the point of view of classification accuracy. To the best of our knowledge, a direct comparison of HMPerceptron and CRFs has not been provided, even though they both were applied to the same Part-Of-Speech tagging problem, with similar results [13,18].

Therefore, on the basis of cited literature, we have chosen the HMPerceptron as our main learning algorithm for the harmonic labeling prediction task. The hypothesis acquired by the HMPerceptron has the form:

$$H(X) = \arg \max_{\bar{Y}=\{\bar{y}_1 \dots \bar{y}_T\}} \sum_t \sum_s w_s \phi_s(X, \bar{y}_t, \bar{y}_{t-1}) \quad (1)$$

where ϕ_s is a *boolean* function of the sequence of events X and of the previous and current labels. The ϕ_s functions are called “features” and allow the algorithm to take into consideration different aspects of the sequence being analyzed. To devise properly the set of features is the “difficult” part in applying the model, since this is the place where both the domain knowledge and the model “simplifications” come to play. The w_s weights are the parameters that

need to be estimated by the learning algorithm. The HMPerceptron applies a simple scheme to do that: it iterates over the training set updating the weights so that features correlated to “correct” outputs receive larger values, and those correlated with “incorrect” ones receive smaller values.

This is actually the same kind of strategy adopted by Rosembat’s perceptron algorithm [20], the only real difference between the two algorithms is in the way the hypothesis is evaluated. In the classification problem faced by the perceptron algorithm, in fact, it is sufficient to enumerate all the possible labels and to pick the best one. In the case of the SSL problem, however, this cannot be done efficiently: the labelling of a single “example” is a sequence of T labels, the number of such labellings is thus exponential in T . To overcome this problem, the algorithm uses Viterbi decoding to pick the best labelling under a first order Markov assumption.

Features Definition. The features we use to model the tonal harmony problem can be arranged into the following classes: *Notes*, *Chord transitions*, *Metric relevance*, *Chord labels* and *Template matching*. In general, features are used to provide discriminative power to the learning system. At each time point t , the features are evaluated in order to compute an informed prediction about the label to associate with the event. The formal definition of the features exploited by the system is provided in Table 1.

Notes features report about the presence/absence of each one of the 12 available pitch classes in each event. For example, the occurrence in the training set of the pitches $\langle 9, 0, 4 \rangle$ (that is $\langle A, C, E \rangle$) associated with the label “A min”, provides evidence that these three pitch classes can be appropriately used to strengthen “A min” label for the event at time t . This information is provided for times t , $t-1$ and $t+1$. So we allow the algorithm to make use of previous and following notes to discriminate among possible chord labels.

Chord transitions feature reports about transitions between chord pairs. The feature allows the HMPerceptron to take into consideration the previous chord label in making the chord prediction about the current event.

Metric relevance features account for the correlation of label changes and the *beat level* of the event being analyzed and of nearby events. In general, the way the meter changes in the neighborhood of a given event is reputed relevant in predicting harmony changes [2]. Then, the metric relevance feature has been devised so to take into consideration the metrical salience of the current and of nearby events as well as the changes in harmony. Since we consider only two possible levels of metric relevance (corresponding to accented and unaccented beats) and adjacent neighbors (at times $t-1$ and $t+1$), we end up with eight possible metrical patterns.

Chord label features report about which label is actually asserted. Intuitively, these features allow the algorithm to evaluate something similar to prior probabilities of the various labels. Here one should consider that there are many possible labels. These features have been devised with the aim at biasing the analysis towards most frequent labels. In principle, these features should be used, other things being equal, to favor frequently stated labels.

Table 1. Feature’s formal definition. Characteristics of the feature vector $x_{.,.}$ are denoted by characteristic-name[$x_{.,.}$]. Three “characteristics” are used in computing the features: “notes” reports the set of pitch classes corresponding to the notes in $x_{.,.}$; “meter” is 1 if event $x_{.,.}$ is accented and 0 otherwise; “tm-chord” reports the chord inferred by the template matching algorithm. Each definition reports the condition that needs to hold for the feature to return 1. Also, each feature reported in the table is actually a feature template which depends on several parameters. The third column clarifies the role of such parameters in each definition. $\bar{y}_t$ and $\bar{y}_{t-1}$ are defined as in Formula (1).

Name	Definition	Parameters
notes	$z \in \text{notes}[x_{.,t+i}] \wedge \bar{y}_t = y'$	z : a pitch class; i : one of -1,0,1 y' : a chord label
chords	$\bar{y}_t = y'$	y' : a chord label
chords transitions	$\bar{y}_{t-1} = y'_1 \wedge \bar{y}_t = y'_2$	y'_1, y'_2 : chord labels
metric relevance	$\text{meter}[x_{.,t-1}] = m_1 \wedge \text{meter}[x_{.,t}] = m_2 \wedge$ $\text{meter}[x_{.,t+1}] = m_3 \wedge \bar{y}_{t-1} = \bar{y}_t$	m_1, m_2, m_3 : 1/0
template matching	$\text{tm-chord}_i[x_{.,t}] = \bar{y}_t$	i : template index

Template matching features report about the chords predicted by a simple template matching algorithm that matches the event against a number of “standard” harmony templates. These features formulate up to five hypotheses for each vertical, abstaining when unsure.

4 Experimental Validation

In order to assess our system, we adopted the Kostka-Payne corpus [8], which is a collection of 45 excerpts from the classical tonal repertoire, ranging from J.S. Bach (1685-1750) to N.D. Ayer (1910-1989), from piano pieces to string Trios and Quartets. B. Pardo annotated the analyses of Kostka and Payne into a set of MIDI files, and made the corpus available on the Net¹. Unfortunately, we were not able to use the full corpus for the experimentation: 11 excerpts could not be used because of problems we encountered in parsing the MIDI files². The final dataset contains a total of 2,622 events in 34 different sequences for an average of 77 events per sequence and a standard deviation of 41 events. The minimum and maximum number of events per sequence are 24 and 208 respectively. There are 73 different chord labels in the whole data set.

Figure 3 reports the performances of two experiments. The first experiment (Figure 3(a)) reports about the average errors made on each musical piece of the corpus. The performances of the system on the training set are reasonably good, they reach an average accuracy of 89.05%, thus demonstrating that the discrimination power of the system is adequate for the problem at hand.

¹ <http://www.cs.northwestern.edu/~pardo>

² We used the parsing routines provided by the standard `javax.sound.midi` package. The full list of the excerpts in the KP corpus is provided in [21].

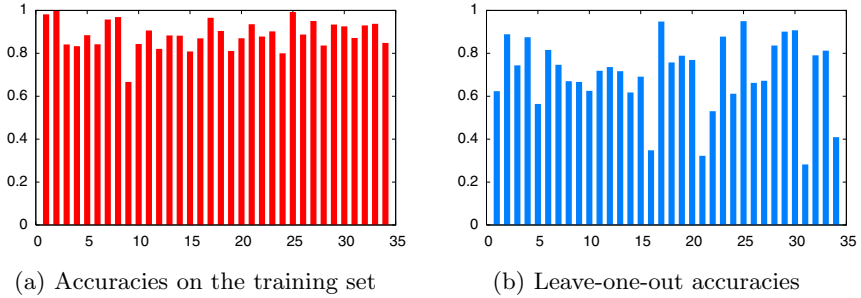


Fig. 3. Accuracies of the BREVE system on the KP corpus

In the second experiment (Figure 3(b)), the system has been iterated 34 times; each time, one of the excerpts has been removed from the training set, and used for testing. This is a leave-one-out cross validation setup, considering each piece one example. The final accuracy of the system is 73.79%.

4.1 Discussion

Looking at the results, we notice that they are accurate but also they overfit the data. This is somewhat surprising, because we were careful in implementing the averaging parameters method suggested in [13]. As reported in that paper, it should guarantee good generalization performances by making the HMPerception work similarly to a very simple ensemble learning algorithm. A possible explanation is that the selected corpus is very challenging from a machine learning perspective. In fact, it encompasses excerpts from authors of very different epochs and styles: some of these appear only once, thus implying that the system is required to generalize from observations encountered in very different excerpts. Also, the above remarks seems to be confirmed by the results of a very similar system that we tested on a much more homogeneous corpus composed only by J.S. Bach chorales: in that case the system achieved about 90% of generalization accuracy [14].

A closer look at our results reveals that most of the errors can be arranged in two main classes, thus providing precious insights to improve the system. A first, obvious, remark is that the sequential information is harder to capture than the information present in the individual events. For instance, there are cases in which the perceptron may have been induced to return such sequence as a result of having being trained over a corpus where romantic language is more represented³. In fact, such cases point out a romantic nuance in the interpretation, that could be corrected should a larger data set be available. A possible way to surmount the problem is to provide the system with some further domain knowledge by adding higher level features. Several errors (namely, 16.34%) are

³ E.g., cases where the sequences have been interpreted in terms of III-II^o-i instead of III-VI-i.

due to the confusion between relative tones: i.e., our output was a major chord, while the correct answer was its relative minor, or vice versa. In a sense, this is itself an encouraging error, indicating that the system provides reasonable output. In the 17.32% of the overall error, we correctly identified the root but not the mode. We registered such problems especially when dealing with arpeggios (e.g., the first movement of the “Moonlight” Sonata Op. 27 n.1 by Beethoven), where harmony is incompletely stated, and it goes back and forth between C $\sharp$ minor and major. In both cases, it would have been crucial to consider features that take into account larger contexts. In several cases, our analysis is somehow more concise (or less detailed) than the “correct” one, disregarding, for instance, passing tones. Sometimes we noted the opposite problem, i.e., BREVE provides analyses that are more detailed. Again, collecting information from a broader neighborhood is likely to help much in those situations. The good news are that enlarging the context should not affect very much the performances. In fact conditional models allow considering arbitrarily distant contextual cues, without high additional costs, as long as they do not require the evaluation of faraway labels.

5 Conclusions

This paper has presented BREVE, a novel system for performing tonal harmony analysis. The main contributions are the following. We developed a system where a HMPerceptron exploits pitch classes in nearby events, metrical accents patterns, chordal successions, as well as suggestions given by a template matching algorithm. The system has been subject to a systematic experimentation and provided encouraging results. Most errors fall into few classes. Their analysis provides meaningful insights about the system behavior.

The analysis problem confirmed to be both interesting and challenging. Part of the difficulties surely comes from the heterogeneity and meagerness of the selected excerpts corpus, but the evidence is that much work is still needed to build systems suitable for tackling the problem in its entirety. Problems come from different directions: known algorithms still require large computational resources; higher level features need to be engineered to improve performances and accuracy; knowledge extraction tools, still beyond the reach of actual systems, would help scientists to explain how music is composed, listened, performed and, ultimately, understood. To develop such higher level features, to augment the corpus of the data, to extract declarative knowledge from the inferred hypotheses, are all interesting directions for future research.

References

1. Bent, I., ed.: Music Analysis in the Nineteenth Century. Cambridge University Press, Cambridge (1994)
2. Lerdahl, F., Jackendoff, R.: A Generative Theory of Tonal Music. MIT Press, Cambridge, MA (1983)

3. Sloboda, J.: *The Musical Mind. The Cognitive Psychology of Music.* Oxford University Press, Oxford, UK (1985)
4. Widmer, G.: *On the Potential of Machine Learning for Music Research.* In: *Readings in Music and Artificial Intelligence.* Harwood Academic Publishers (2000)
5. Lopes de Mantaras, R., Arcos, J.: *AI and Music: From Composition to Expressive Performance.* *AI Magazine* **23** (2002) 43–57
6. Temperley, D.: *The Cognition of Basic Musical Structures.* MIT, Cambridge, MASS (2001)
7. Pardo, B., Birmingham, W.P.: *Algorithms for Chordal Analysis.* *Computer Music Journal* **26** (2002) 27–49
8. Kostka, S., Payne, D.: *Tonal Harmony.* Number New York. McGraw-Hill (1984)
9. Schoenberg, A.: *Structural Functions in Harmony.* Norton, New York (1969)
10. Ebcioğlu, K.: *An Expert System for Harmonizing Chorales in the Style of J. S. Bach.* *Journal of Logic Programming* **8**(1) (1990) 145–185
11. Winograd, T.: *Linguistics and Computer Analysis of Tonal Harmony.* *Journal of Music Theory* **12** (1968) 2–49
12. Raphael, C., Stoddard, J.: *Functional Harmonic Analysis Using Probabilistic Models.* *Computer Music Journal* **28**(3) (2004) 45–52
13. Collins, M.: *Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms.* In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* (2002)
14. Radicioni, D.P., Esposito, R.: *Learning Tonal Harmony from Bach Chorales.* In Fum, D., del Missier, F., Stocco, A., eds.: *Procs. of the 7th International Conference on Cognitive Modelling.* (2006)
15. Dietterich, T. In: *Machine Learning for Sequential Data: A Review.* Volume 2396 of *Lecture Notes in Computer Science.* Springer-Verlag (2002) 15–30
16. Dietterich, T.G., Ashenfelder, A., Bulatov, Y.: *Training Conditional Random Fields via Gradient Tree Boosting.* In: *Procs. of the 21st International Conference on Machine Learning,* New York, NY, USA, ACM Press (2004)
17. McCallum, A., Freitag, D., Pereira, F.: *Maximum Entropy Markov Models for Information Extraction and Segmentation.* In: *Procs. of the 17th International Conference on Machine Learning,* Morgan Kaufmann, San Francisco, CA (2000) 591–598
18. Lafferty, J., McCallum, A., Pereira, F.: *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.* In: *Procs. of the 18th International Conference on Machine Learning,* San Francisco, CA (2001) 282–289
19. Altun, Y., Hofmann, T., Smola, A.J.: *Gaussian Process Classification for Segmenting and Annotating Sequences.* In Greiner, R., Schuurmans, D., eds.: *Proceedings of the Twenty-first International Conference on Machine Learning,* Banff, Canada, ACM Press, New York, NY (2004)
20. Rosenblatt, F.: *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.* *Psychological Review* (Reprinted in *Neurocomputing* (MIT Press, 1998)) **65** (1958) 386–408
21. Radicioni, D.P., Esposito, R.: *Sequential Learning for Tonal Analysis.* Technical Report RT 92/05, Università degli Studi di Torino (December 2005)

Hypothesis Diversity in Ensemble Classification

Lorenza Saitta

Università del Piemonte Orientale, Dipartimento di Informatica
Via Bellini 25/G, Alessandria, Italy

Abstract. The paper discusses the issue of hypothesis diversity in ensemble classifiers. The measures of diversity previously proposed in the literature are analyzed inside a unifying framework based on Monte Carlo stochastic algorithms. The paper shows that no measure is useful to predict ensemble performance, because all of them have only a very loose relation with the expected accuracy of the classifier.

1 Introduction

In ensemble classification and learning it is widely acknowledged that diversity among component hypotheses is beneficial to the classifier's performances [1]. Reasonable as this claim may be, in practice it has not been possible, up to now, to define a *diversity* measure clearly linked to the classifier's accuracy. Kuncheva and Whitaker [6] have presented an extensive overview of diversity measures; an experimental analysis of their properties led Kuncheva [5] to a pessimistic view of their usefulness.

In this paper we analytically analyze the various notions of diversity described by Kuncheva and Whitaker [6], by setting them into a unifying framework, previously proposed by Esposito and Saitta [3] [2], which exploits the same *oracle output* representation as used by those authors.

The conclusions of the study support Kuncheva's negative view, and show, in addition, that not only no useful measure exists today, but that it is unlikely that one will ever exist. In fact, on the one hand, the link between diversity and performance is not monotonic (namely, the greatest diversity does not correspond to the best performance), and, on the other, the level of diversity required to achieve the best performance for a given number T of component hypotheses depends, in a non-monotonic way, upon T itself. In other words, the best overlap among T hypotheses on a given set of examples is not the best one if a further hypothesis is added. Then, the (dis)agreements among hypotheses must be re-distributed over the examples. It is clear that this lack of monotonicity greatly reduces the hopes to find a useful definition of diversity.

The remaining of the paper is organized as follows: Section 2 introduces the unifying framework used in the analysis. Section 3 presents the analysis of the diversity measures based on *local* diversity (averages of the diversity between pairs of hypotheses), whereas Section 4 discusses the *global* measures. Section 5 contains some conclusions.

2 Problem Setting

For the sake of simplicity, we only consider in this paper binary classification problems, i.e., $Y = \{+1, -1\}$. Let X be a finite set of examples, ω the target concept, and WL a (weak) learner. We will have $\omega(x) = +1$ for any x belonging to the concept (positive example), and $\omega(x) = -1$ for any x not belonging to the concept (negative examples). Let $|X| = N$ and let $\mathbf{d}$ be a fixed probability distribution over X .

Let us now consider a set Φ of hypotheses, and let $\mathbf{q}$ be a probability distribution over Φ . In order to provide a synthetic representation of a particular learning problem, an *oracle output* matrix M is introduced by Kuncheva and Whitaker [6]. In this paper we exploit the same framework as Kuncheva [5], i.e., a data-independent, oracle-based set up. In other words, we consider the same matrix M , whose R columns correspond to a set of hypotheses, provided independently of any data examination, and whose N rows correspond to the examples in X . For each example x_k and each hypothesis φ_j we only know whether φ_j is correct on x_k (entry $p_j(x_k) = 1$) or wrong on it (entry $p_j(x_k) = 0$).

Starting from M , we can compute a number of useful quantities. The average taken w.r.t. column j ,

$$r_j = \sum_{k=1}^N d_k p_j(x_k)$$

denotes the probability that an x_k , extracted from X according to $\mathbf{d}$, is correctly classified by a given φ_j ; in other words, the averages taken on the columns correspond to the accuracies r_j of the single hypotheses in Φ , usually considered in Machine Learning. We collect the r_j 's into a vector $\mathbf{r}^T = \mathbf{d}^T M$ of length R .

In an analogous way, the average taken along the rows of M , i.e.,

$$p(x_k) = \sum_{j=1}^R q_j p_j(x_k)$$

denotes the probability that a hypothesis φ_j , extracted from Φ according to $\mathbf{q}$, correctly classifies a given example x_k . In other words, by choosing T hypotheses from Φ , on the average $T p(x_k)$ of them will correctly classify x_k . Again, we collect the $p(x_k)$ values into a vector $\mathbf{p} = M \mathbf{q}$, with $|\mathbf{p}| = N$.

Given the matrix M , we can compute the overall mean of the values in the matrix:

$$r = \sum_{k=1}^N \sum_{j=1}^R d_k q_j p_j(x_k) \quad (1)$$

The vector $\mathbf{r}$ contains the important parameters for hypothesis selection. On the other hand, the vector $\mathbf{p}$ contains those for hypothesis combination. The vectors $\mathbf{p}$ and $\mathbf{r}$ are only loosely related through relation (1).

In order to exploit the internal structure of the matrix M , we reorder rows and columns in such a way that the $p(x_k)$ values are decreasing from top to bottom, whereas the r_j values are decreasing from left to right. This reordering only amounts to renaming both examples and hypotheses.

Using the hypotheses in Φ , each with the associated multiplicity q_j , we can form a combined classifier:

$$H_R(.) = \text{Sign}(\sum_{j=1}^R q_j \varphi_j(.)) \quad (2)$$

The classifier $H_R(.)$ is correct on example x_k iff $p(x_k) > 1/2$, and incorrect otherwise. Let ρ_R be the accuracy of $H_R(.)$. In the following of the paper we will try to relate ρ_R with the measures of diversity. Moreover, for the sake of simplicity, we will set $d_k = \frac{1}{N}$ for all $k \in [1, N]$, and $q_j = \frac{1}{R}$ for all $j \in [1, R]$; this assumption is not reductive of generality because duplicate examples and hypotheses may be considered in X and Φ , respectively. In this setting, the value ρ_R is simply the cardinality of the subset of X containing those examples for which $p(x_k) > 1/2$.

3 Local Diversity Measures

In this section we consider diversity measures based on pairs of hypotheses in Φ .

3.1 Yule's Q statistic

The Q_{ij} statistics is defined on a pair of hypotheses, φ_i and φ_j . Let $N_{\alpha\beta}(i, j)$, with $\alpha, \beta \in \{0, 1\}$, be the number of examples for which $p_i(x_k) = \alpha$ and $p_j(x_k) = \beta$. Then:

$$Q_{ij} = \frac{N_{11}(i, j)N_{00}(i, j) - N_{01}(i, j)N_{10}(i, j)}{N_{11}(i, j)N_{00}(i, j) + N_{01}(i, j)N_{10}(i, j)} \quad (3)$$

In our notation, we have:

$$N_{11}(i, j) = \sum_{k=1}^N p_i(x_k) \cdot p_j(x_k) \quad (4)$$

Then:

$$N_{00}(i, j) = \sum_{k=1}^N (1 - p_i(x_k)) \cdot (1 - p_j(x_k)) \quad (5)$$

$$N_{10}(i, j) = \sum_{k=1}^N p_i(x_k) \cdot (1 - p_j(x_k)) \quad (6)$$

$$N_{01}(i, j) = \sum_{k=1}^N (1 - p_i(x_k)) \cdot p_j(x_k) \quad (7)$$

$$(8)$$

Expression (3) is averaged over all the pairs $\varphi_i, \varphi_j \in \Phi$, obtaining thus Q . Even though a Q value close to 1 should not be favorable to ensemble classification, whereas a Q value close to -1 should, it is easy to find counterexamples. For instance, by considering three hypotheses, each one with individual accuracy 0.50, one may obtain the following values: $(Q = 1, \rho_R = 0.5)$, $(Q = -0.33, \rho_R = 0.5)$, $(Q = 0, \rho_R = 0.75)$. On the other hand, with three hypotheses, each with individual accuracy 0.80, one may obtain the following values: $(Q = 1, \rho_R = 0.8)$, $(Q = -1, \rho_R = 1)$ and $(Q = -0.057, \rho_R = 0.8)$. In this case, the behavior of ρ_R vs. Q is the opposite than in the previous case.

3.2 Disagreement Measure Dis

This measure uses the same base values $N_{\alpha\beta}(i, j)$ as Yule's Q statistic, but computes the following pairwise value:

$$Dis_{ij} = \frac{N_{01}(i, j) + N_{10}(i, j)}{N} \quad (9)$$

It is easy to show that:

$$Dis_{ij} = \frac{1}{N} \sum_{k=1}^N [p_i(x_k) + p_j(x_k) - 2 p_i(x_k) p_j(x_k)] \quad (10)$$

If we take the average of Dis_{ij} over all pairs of hypotheses, we obtain:

$$Dis = \frac{2}{R(R-1)} \sum_{i=1}^{R-1} \sum_{j=i+1}^R Dis_{ij} = \frac{2R}{(R-1)N} \sum_{k=1}^N p(x_k)[1 - p(x_k)] \quad (11)$$

We obtain $Dis = 0$ iff all the $p(x_k)$'s are either 1 or 0. In this case, the matrix M is partitioned into an upper part with $p_j(x_k) = 1$ for $j \in [1, R]$ and $k \in [1, S]$ for some S , and a lower part, consisting of $(N-S)$ rows with all entries equal to 0. The theory underlying the disagreement measure says that $Dis = 0$ is bad for ensemble classification; in fact, a value $Dis = 0$ corresponds to a case in which the asymptotic accuracy cannot be better than the best accuracy of the single hypotheses. On the other hand, things become more fuzzy when Dis increases.

Let us suppose that all the $p(x_k)$ values are a little greater than 0.5, i.e., $p(x_k) = 1/2 + \epsilon$. In this case, $Dis \simeq \frac{R}{2(R-1)}$, which is close to the maximum value. In this case $\rho_R = 1$. On the other hand, if all the $p(x_k)$ have a value a little lower than 0.5, i.e., $p(x_k) = 1/2 - \epsilon$, Dis has the same value as before, which is again very close to the maximum. In this case, however, $\rho_R = 0$. When Dis increases, the relation between Dis and ρ_R become looser and looser, and thus less and less useful.

3.3 Double-Faults Measure DF

This measure only uses N and $N_{00}(i, j)$. Specifically, we can easily show that:

$$DF_{ij} = \frac{1}{N} \sum_{k=1}^N [1 - p_i(x_k)] [1 - p_j(x_k)] \quad (12)$$

and the average value will be:

$$DF = (1 - r) - r \frac{R}{R - 1} + \frac{R}{N(R - 1)} \sum_{k=1}^N p^2(x_k) \quad (13)$$

DF takes values in the interval $[0, 1]$, and is good for ensemble classification when it is high. In fact, if all $p(x_k)$ are close to 0, $DF \simeq 1$, which implies $r \simeq 0$ and $\rho_R \simeq 0$. On the other hand, if all $p(x_k)$ are close to 1, $DF \simeq 0$, which implies $r \simeq 1$ and $\rho_R \simeq 1$. In these two cases, even though intuitively correct, DF does not bring any new information with respect to the knowledge of r .

4 Global Diversity Measures

In this section we consider the diversity measures based on properties of all the hypotheses in Φ .

4.1 Entropy Measure

The entropy measure E is the first of the diversity measures which is global, i.e., it is not an average of pairwise diversities. Using our notation, we can write E as follows:

$$E = \frac{R}{N(R - \lceil R/2 \rceil)} \sum_{k=1}^N \min[p(x_k), 1 - p(x_k)] \quad (14)$$

Values of E close to 0 indicates a disfavourable situation for ensemble classification. In fact, when $E \simeq 0$, all $p(x_k)$ values are either 1 or 0, and ρ_R is not better than the individual accuracies. When E is close to 1, most $p(x_k)$ are close to $1/2$. Let, in this case, be S the number of examples with $p(x_k) = 1/2 + \epsilon$ and $(N-S)$ the number of examples with $p(x_k) = 1/2 - \epsilon$. With the same value $E \simeq 1$ we may have both $\rho_R \simeq 0$ and $\rho_R \simeq 1$.

4.2 Kohavi-Wolpert Variance

This measure of diversity can be written as:

$$KW = \frac{1}{N} \sum_{k=1}^N p(x_k)[1 - p(x_k)] = \frac{R - 1}{2R} Dis \quad (15)$$

As KW is proportional to Dis , both measures show the same behaviour.

4.3 Interrater Agreement

This measure, also called κ *measure*, makes use of the average global accuracy r . Specifically:

$$\kappa = 1 - \frac{R}{(R-1)Nr(1-r)} \sum_{k=1}^N p(x_k)[1-p(x_k)] = 1 - \frac{1}{2r(1-r)} Dis \quad (16)$$

For any given r , the κ measure behaves as the complement to 1 of Dis .

4.4 Measure of "difficulty" θ

This measure derives from the work of Hansen and Salamon [4], and is based on the distribution of the $p(x_k)$ values. Actually, as $p(x_k) = \frac{1+\mu(x_k)}{2}$, where $\mu(x_k)$ is the margin of example x_k (see [3] for details), the measure of difficulty θ is related to the distribution of the margins. In our notation, we can write θ as follows:

$$\theta = \frac{4}{N} \sum_{k=1}^N p^2(x_k) - 4r^2 \quad (17)$$

By approximating the ratio $R/(R-1)$ with 1, for the sake of simplicity, the measure θ can be related to the preceding measures as follows:

$$\begin{aligned} \theta &= 4[DF - (1-r)^2] \\ \theta &= 4r(1-r) - \frac{2}{N} Dis = \frac{4}{N} KW \\ \theta &= 4r - (1-\kappa)r(1-r) \end{aligned} \quad (18)$$

As θ is related only through r to the other measures, for any given r its behaviour can be deduce from their's.

4.5 Generalized Diversity

This measure, called GD , varies between 0 (minimum diversity) and 1 (maximum diversity). In our terms:

$$GD = 1 - \frac{RN(1-2r) - N}{(1-r)(R-1)N} - \frac{R}{(1-r)(R-1)N} \sum_{k=1}^N p^2(x_k) \quad (19)$$

The value of GD can be related to that of θ .

4.6 Coincident Failure Diversity

The last measure considered by Kuncheva and Whitaker [6] has been introduced by Partridge and Krzanowski [7]. By denoting by p_0 the probability that all classifiers are always correct, i.e., all entries in the matrix are equal to 1, the

Coincident failure diversity (CFD) is equal to 0 (minimum value) when $p_0 = 1$, and reaches its maximum value, 1, when at most one classifier will fail on any randomly chosen example. Then:

$$\begin{aligned} CFD &= 0 && \text{if } p_0 = 1 \\ CFD &= \frac{R}{R-1} \frac{r}{1-p_0} && \text{otherwise} \end{aligned} \quad (20)$$

As a conclusion from the above analysis, we can say that it does not seem possible to precisely define a diversity measure over a set of hypotheses which captures a reliable relation with the accuracy of the ensemble classifier. Experimental results reported by Kuncheva and Whitaker [6] provide empirical support to this claim.

A comment can be done on the observation that "*The intuition that less accurate classifiers will form more diverse ensemble is not supported by this experiment*" [6]. Actually, from a theoretical point of view, it is true that less accurate hypotheses MAY produce more diverse classifiers, but this is not always a positive thing. In fact, if we combine three classifiers, all with individual accuracies $r = 0.30$, the three hypotheses can correctly classify totally disjoint example sets; but, in this case, the ensemble accuracy will be 0. It is clear that, considering two hypotheses φ_1 and φ_2 , with individual accuracies r_1 and r_2 both greater than $1/2$, the hypotheses must agree on at least a fraction $|r_1 - r_2|$ of the examples. Hence, lower individual accuracies may imply less overlap.

5 Gorillas and Their Heights

In a further study, Kuncheva [5] takes inspiration from previous biological studies on diversity [8]. In life sciences Rao [8] introduced a general definition of diversity, applied to *populations*. Given a set $\mathcal{P}$ of probability distributions over a given population, a distance measure $H(P)$ is any non-negative and concave function of P . Then, given any distance measure $\zeta(x_1, x_2)$ and two elements P_1 and P_2 in $\mathcal{P}$, their *dissimilarity* D_{ij} is defined by:

$$D_{ij} = H(P_1, P_2) - \frac{1}{2} [H(P_1) + H(P_2)] \quad (21)$$

where $H(P_1)$ and $H(P_2)$ are the average *intra-population* dissimilarities and $H(P_1, P_2)$ is the *inter-population* dissimilarity.

A problem that Kuncheva debates in the paper is whether we shall consider as *populations*, in Rao's theory, the gorillas or their heights. Both cases are then considered. In the ensemble classification terms, considering as populations the rows of the matrix means to evaluate how the hypothesis set as a whole behave on any two different examples; considering as populations the columns of the matrix means to evaluate how any two hypotheses differ in behavior on the whole set of examples.

Rows as populations - Let us consider one example x_k and the whole set Φ of hypotheses. In order to compute the dissimilarities within x_k , and across

x_k and x_h , we need to define a distance measure. To this aim, let us consider the following ζ_k function:

$$\zeta_k(\varphi_i(x_k), \varphi_j(x_k)) = \begin{cases} 1 & \text{if } p_i(x_k) \neq p_j(x_k) \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

As the function ζ_k is symmetric with respect to its arguments, and $\zeta_k(\varphi_i, \varphi_i) = 0$ for all x_k , the intra-population dissimilarity can be computed as follows:

$$H(x_k) = \frac{1}{2} \sum_{i=1}^T \sum_{j=1}^T q_i q_j I_{p_i(x_k) \neq p_j(x_k)} \quad (23)$$

By noticing that we have:

$$I_{p_i(x_k) \neq p_j(x_k)} = [p_i(x_k) - p_j(x_k)]^2 \quad (24)$$

we obtain, after some elaborations:

$$H(x_k) = p(x_k)[1 - p(x_k)] \quad (25)$$

Let us consider now another example x_h :

$$H(x_h) = p(x_h)[1 - p(x_h)] \quad (26)$$

Moreover:

$$H(x_k, x_h) = \frac{1}{2} \sum_{i=1}^T \sum_{j=1}^T q_i q_j [p_i(x_k) - p_j(x_h)]^2 = \frac{1}{2} [p(x_k) + p(x_h) - 2p(x_k)p(x_h)] \quad (27)$$

Finally, the dissimilarity between two examples x_k and x_h is evaluated as follows:

$$D_{kh} = \frac{1}{2} [p(x_k) + p(x_h) - 2p(x_k)p(x_h)] - \frac{1}{2} p(x_k)[1 - p(x_k)] - \frac{1}{2} p(x_h)[1 - p(x_h)] \quad (28)$$

$$D_{kh} = \frac{1}{2} [p(x_k) - p(x_h)]^2 \quad (29)$$

Columns as populations - Let us consider now two columns and repeat the computation above. The result is that the dissimilarity computed with columns as populations coincides with the disagreement measure Dis , and, then, its behaviour is the same.

6 Conclusions

In this paper we have shown that diversity measures among hypotheses can be described and related each other within a unifying scheme; this scheme allows the asymptotic error of a majority voting combination of R hypotheses to be exactly computed. In this way it is easy to let the links between alternative measures to emerge.

The conclusions of this work support the view that no current measure really captures a notion of diversity useful for ensemble classification. The only measure directly correlated with the ensemble error is the ratio between the cardinality of the set of examples with $p(x_k) > 1/2$ and N . In the data-independent setting of this paper, this ratio IS the error, and then this definition appear to be tautological. On the other hand, if we consider a learning set and a test set, the ratio, estimated on the learning set, can be used as an estimate of the diversity among the set of learned hypotheses, and it does not coincide with the error.

References

1. T. Dietterich. Machine learning research: Four current directions, 1997.
2. Roberto Esposito and Lorenza Saitta. A monte carlo analysis of ensemble classification. In Russ Greiner and Dale Schuurmans, editors, *Proceedings of the twenty-first International Conference on Machine Learning*, pages 265–272, Banff, Canada, July 2004. ACM Press, New York, NY.
3. Roberto Esposito and Lorenza Saitta. Experimental Comparison between Bagging and Monte Carlo Ensemble Classification. In Luc De Raedt and Stefan Wrobel, editors, *Proceedings of the twenty-second International Conference of Machine Learning*, pages 209–216, Bonn, Germany, August 2005. ACM Press, New York, NY.
4. Lars Kai Hansen and Peter Salamon. Neural networks ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, October 1990.
5. L. Kuncheva. That elusive diversity in classifier ensembles, 2003.
6. Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51:181–207, 2003.
7. D. Partridge and W. Krzanowski. Distinct failure diversity in multiversion software, 1997.
8. C. R. Rao. Diversity: Its measurement, decomposition, apportionment and analysis. *The Indian Journal of Statistics, Series A*, 44:1–22, 1982.

Complex Adaptive Systems: Using a Free-Market Simulation to Estimate Attribute Relevance

Christopher N. Eichelberger and Mirsad Hadžikadić

The University of North Carolina at Charlotte, College of Information Technology,
9201 University City Boulevard, Charlotte, NC 28223-0001
{ceichelb, mirsad}@uncc.edu

Abstract. The authors have implemented a complex adaptive simulation of an agent-based exchange to estimate the relative importance of attributes in a data set. This simulation uses an individual, transaction-based voting mechanism to help the system estimate the importance of each variable at the system/aggregate level. Two variations of information gain – one using entropy and one using similarity – were used to demonstrate that the resulting estimates can be computed using a smaller subset of the data and greater accommodation for missing and erroneous data than traditional methods.

1 Introduction

Complex adaptive systems (CAS) as described in [6] are generally goal-driven, multi-agent systems in which one or more of the agents adapt over time. The patterns that emerge can be useful in solving difficult problems [2][1][3]. Estimating the relevance of attributes in an arbitrary data set is a difficult problem, and one which the authors set out to address using a CAS model of a free-market exchange. Common analysis methods such as C4.5 [10] and factor analysis [5] are designed for well-defined and reasonably-constrained problems, such as “What single variable best correlates with personal income?” They are much less useful when exploring large data sets that have many columns and no specific research question.

In the following sections, the authors present a CAS method for estimating the relevance of features in an arbitrary data set; and the preliminary results on a sample data set.

2 Method

The CAS simulation consists of an environment populated by agents. Each of these entities is described here in greater detail, followed by a brief discussion of the main simulation event loop.

2.1 Environment

The primary purpose of the environment in this CAS simulation is to contain the agents and the data to be analyzed. Though we have tested the simulation harness with a handfull of different data sets – including a few data sets from medicine and finance for which traditional statistical approaches such as factor analysis and principal component analysis happen not to have run successfully – most of the results included in this treatment concern the mushrooms dataset from the UCI Machine Learning repository [9].

The environment also provides functions that report the information gain [10] that result from partitioning a collection of instances based on the values of a single attribute. Information gain seeks minimize the disorder among partitions, but is susceptible to exploitation: If an attribute has exactly one value per instance (such as a unique record identifier), then that attribute will partition the records into entirely uniform groups (one record per group), but without adding any benefit to the analysis. A fairly standard correction (the gain ratio, also described in [10]) was applied to offset this effect.

Notice that this description of information gain considered disorder (non-uniformity) in terms of a single outcome attribute. This is very useful in many applications, and agrees with the traditional definition of information gain in terms of information entropy. There are cases - when the analysis task is not well defined, or when multiple analysis tasks are to be undertaken with the same data set - that it may be more useful to define the disorder or non-uniformity in terms of all of the features rather than the value of a single attribute in the collection. To accomplish this, an alternate form of information gain was introduced that uses a similarity measure to quantify the change in non-uniformity resulting from partitions built based on a single-attribute split.

Previous work done by one of the authors [4] defines similarity as equation (1), a linear combination of the number of common and unique features between any pair of instances. The definition of $\rho(A, B)$, the common features between the two instances, and $\delta(A, B)$, the differences between the two instances accounts for the symmetry of the similarity function.

$$s(A, B) = s(B, A) = \frac{\rho(A, B) - \delta(A, B)}{\rho(A, B) + \delta(A, B)} \quad (1)$$

$$\rho(A, B) = |A \cap B|, \quad \delta(A, B) = |(A - B) \cup (B - A)|$$

This formula works well for pair-wise similarity, but introduces too many computations, on the order of $O(n^2)$, to be useful in a CAS simulation in which the function must be called repeatedly for every agent. Instead, the authors have substituted a simplified version of similarity that operates on the collection of instances as one unit, estimating the internal uniformity of the collection as per equation (2). Notice that, for similarity to be used as an alternate to entropy in computing information gain, $1 - \textit{similarity}$ is reported as the disorder (dissimilarity) of the system.

$$s(A, B) = \frac{\sum_a (N(a) \cdot M(a))}{\sum_a N(a)}$$

$$N(a) = \sum_{f \in (A \cup B)} \begin{cases} |f| & \text{iff } f_a = a \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$M(a) = \max_{f \in (A \cup B)} \begin{cases} P(f) & \text{iff } f_a = a \\ 0 & \text{otherwise} \end{cases}$$

wherein f is a feature, or attribute-value pair; $|f|$ is the number of copies of the feature f ; $P(f)$ is the probability (or density) of feature f ; and f_a denotes the attribute of feature f .

The trade-offs between the two methods, entropy and similarity, are reasonably obvious: entropy is a very goal-specific method, relatively inured to noise in the data set; similarity is a goal-agnostic method, susceptible to noise, but able to perform well when the majority of columns are meaningful. To help illustrate how these two methods are related, they were both used to estimate the gain provided by each attribute in the UCI mushrooms set, and are reported in the Figure 1. Note that these tests were run using these two methods outside of the CAS. This was done to establish a baseline for comparison against the CAS results.

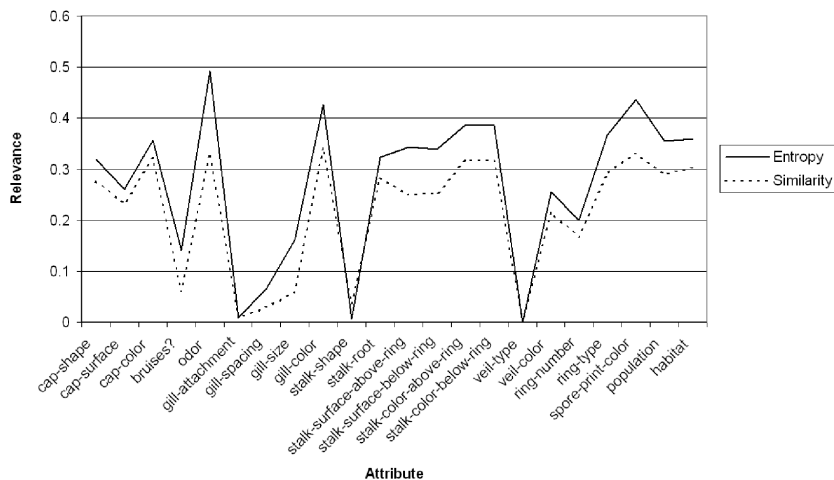


Fig. 1. Non-CAS entropy vs. similarity, full data

There are variations between the two curves, but the average rank of attributes gain evinces reasonable agreement, suggesting that the two methods, while different in focus, are both reasonable choices for estimating gain. This result is consistent with our observations running the system on other data sets. These two methods were contrasted further in the CAS test runs; see Results and Discussion, below.

2.2
Agents

Each agent in this CAS simulation represents one complete instance (record) from the data set, and is charged with identifying the most relevant attributes in the entire data set. It does so by computing the information gain for each of the attributes of which it has knowledge, and using these gain values as price points for buying and selling features with its peers. See Figure 2. Each agent begins the simulation with the following: a set of observations initialized from its own properties in the data set; an estimate of the gain each attribute provides, based solely on the agents own observations; \$1.00 in currency to spend on features from other agents; and a purchasing strategy.

Consider this initial state: When an agents observations consist of only its own features, then any attribute can produce no gain under either method, entropy or similarity. This means that all agents begin by valuing all attributes equally. It is only after an agent has collected (purchased) features from other agents that its estimate of information gain per attribute can diversify.

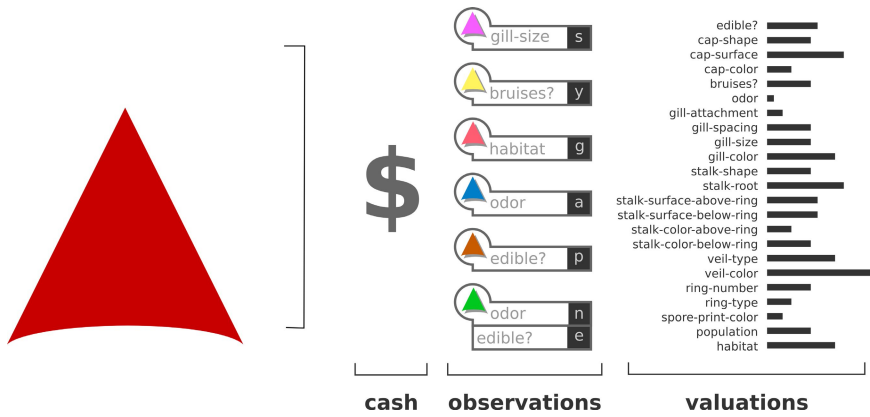


Fig. 2. CAS agent

The purchasing strategy is the same for all agents. Each turn:

1. The agent first updates its valuation of the attributes based on the observations (features) the agent has accumulated so far:
 - (a) Each attributes gain becomes its value to that agent; that is, the gain becomes the price for which this agent will sell features having this attribute and the price for which this agent will buy features having this attribute.
2. When purchasing, it examines its visible peers (those within a local cone in front of the agent), and finds the peer that has the best feature (that feature for which the buyers valuation is maximal) for sale at a price it can afford.
3. If it finds any such feature, it makes a purchase. In the event that there are multiple sellers or features that qualify, one of the candidates is selected randomly.

- (a) the purchaser accumulates one additional observation, an attribute/value pair tagged with the sellers ID
 - (b) the purchaser transfers the sales price from its own cash reserves to the sellers
 - (c) a message about the purchase is also sent to the environment
4. After having made a purchase (or not) the agent seeks to move away from the agents from whom it has most recently collected observations. This is merely to encourage the agents to seek novelty in its purchases. Agents are prevented from purchasing the same observation twice from the same seller, though it is permissible for the same observation to be purchased from different sellers.

The default agent purchasing strategy generally equates to a search for counter-evidence: If an agent values attribute a_i most highly, then it seeks to purchase other features of the form $(a_i, v_?)$. If, after having incorporated the new features into the collection of observations, and having re-evaluated the relative importance of each attribute, the new features reinforce the evaluation that a_i is the most valuable attribute, the agent will continue to try to purchase even more features with this attribute. When this attribute fails to report competitive information gain, the agent will begin trying to purchase features that use another attribute. Because the agents possess different sets of observations and have different neighbors, their estimates of the relative worth (gain) of attributes will differ even though they all use the same learning method. One of the authors' research goals was to identify the degree to which these local decisions would aggregate to attribute relevance values that were comparable to the non-CAS methods.

2.3 Simulation Event Loop

To examine the ability of the CAS simulation to draw global conclusions from local data, the authors have adopted a windowed approach to which records are allowed to interact as agents in the simulation. In this scheme, a random subset of records are deployed as active agents in the exchange. The agents are allowed to interact for some number of time-steps at each time step the agents re-evaluate the gain of attributes (prices) based on their individual observations, and then search for a suitable purchase their purchases are recorded centrally, both the number of times each attribute was purchased and the price for which it was purchased, the output displays are updated, and then the subgroup is dissolved, and is replaced by a new subgroup until the maximum number of subgroups (defined by the user) has been run.

When the subgroups have all been run, the system reports the final attribute relevance as the average purchase price per attribute.

3 Results and Discussion

The experiments reported here represent 20 subgroups per run, 200 time-steps per subgroup, 100 agents per subgroup. For the first series, traditional

entropy-based information gain was used; in the second series, similarity-based information gain was used; lastly, the two gain methods are contrasted. No single simulation run ever mixed the two uniformity measures.

3.1 Contrasting Entropy and Similarity

Superficially, the salient difference between the entropy-based gain and the similarity-based gain is that the former is dependent upon the outcome variable, while the latter is not. But the effects of these two methods are not so obvious when observed in the test runs.

There are two major comparisons to make between the entropy-based gain and the similarity-based gain: how they affect the non-CAS simulation under increasingly sparse data; and how they affect the CAS runs.

Contrasting Entropy and Similarity in a Non-CAS Run. To estimate the performance of the two gain methods under different quantities of missing data, the authors ran two major tests: one with only 1% data knocked out, and one with 99% of the data knocked out. The results are reported in Figure 3.

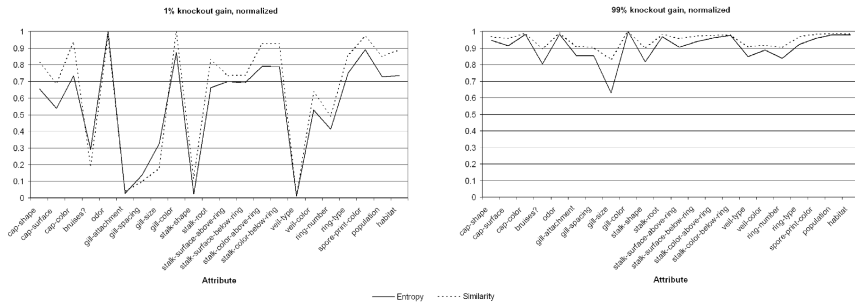


Fig. 3. Non-CAS entropy vs. similarity at 1%, 99% knockout

Note that – even though the attribute relevance values have been normalized – the two gain methods agree reasonably well in both cases. The other obvious trend is that the ability of both methods to discriminate among the attributes is diminished as the volume of missing data increases. To observe this effect more directly, a series of tests was conducted (again, outside of the CAS). The results are reported in Figure 4 for the entropy-based and similarity-based methods.

These scatter plots are constructed as follows: The x -axis is the normalized relevance of the attributes as computed non-CAS for the full data set. The y -axis is the normalized relevance reported by each test run (series). Hence, the points from the full-data test all fall on the identity line. Other runs do not fall on the identity line, and the distance from the identity line reflects how much the attribute relevance drifts from the full-data series as more data are knocked out of the test sets.

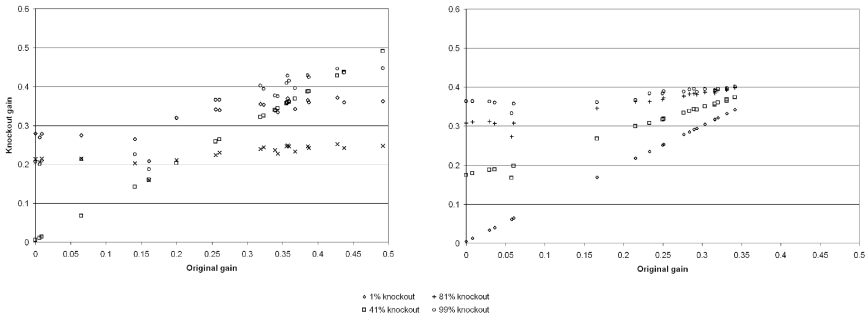


Fig. 4. Entropy’s and similarity’s responses to knockout

It is interesting to observe that the two methods do not respond to an increasing amount of missing data in the same way. The similarity-based gain has a much smoother transition from full data to missing data, whereas the entropy-based gain exhibits a much more tumultuous transition. The authors conclude that the entropy-based gain is much more sensitive to missing data than is the similarity-based gain; this is sensible, since the entropy measure relies on knowing the value of the outcome variable. If that value is missing, entropy is sure to misreport the power of a partitioning attribute.

Contrasting Entropy and Similarity in CAS. The CAS estimates of attribute relevance for the UCI mushrooms data set are reported in Figure 5.

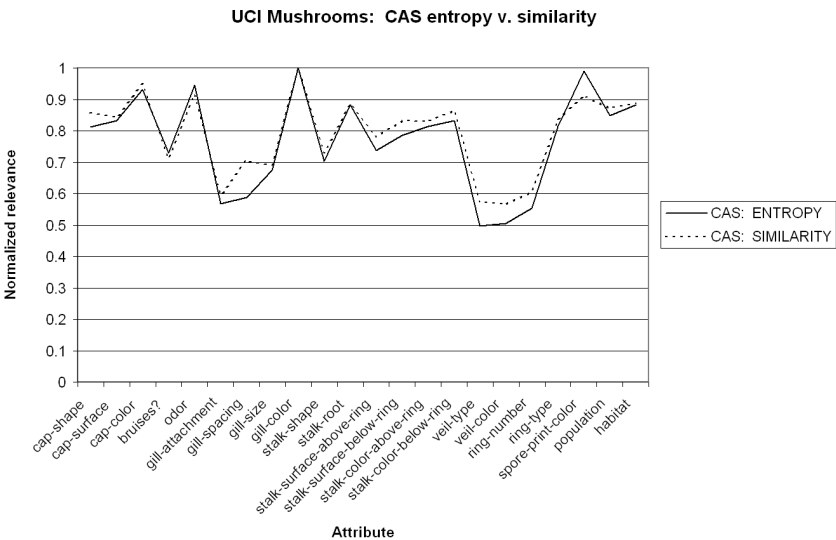


Fig. 5. CAS entropy vs. similarity

These two methods report remarkably similar results, suggesting that many of the variables are correlated with the target outcome variable (whether the mushroom is poisonous or edible). If this were not the case, then there should be more disagreement between the two methods. And yet, closer consideration reveals that the two methods' agreement is not necessarily straight forward. Consider:

The entropy-driven gain computes uniformity based only on the value each instance has for only the target attribute. If very few instances target attribute-values are known to an agent, then any partitioning scheme will lump those unknowns together as unknown-target-attribute value instances. This happens both for the entire set of observations (the baseline entropy) as well as in the partitions, but because the instances lacking outcomes always agree with each other (that is, they never diminish each others uniformity), the gain is effectively computed using only the instances for which the outcome value is known. This explains why the off-line estimate of the top-level gain for attributes agrees reasonably well with the gain reported by the price-per-attribute in the CAS simulation.

The similarity-driven gain computes uniformity based on the values each instance has for all attributes. Every observed feature drives similarity, but when observations are sparse as is generally the case for the agents in the CAS simulation this effect is magnified, and is likely to reward inordinately those attributes that have a high number of unique values, since those partitions are likely to contain features from only one record in an agents experience.

Entropy-driven agents consider missing features to contribute to the homogeneity of subgroups, while similarity-driven agents consider missing features to contribute to the heterogeneity of the same subgroups. And yet, despite these effects, the interaction of local evaluations based on woefully limited subsets of sparse data are sufficient to extract a high degree of useful information about the relative importance of the attributes at the global level.

Emergence. The “complexity” in a complex adaptive system generally refers to the observation that patterns emerge over the course of the run that are more complex than what might be expected from an inspection of the CAS program. The problem with this definition is that it uses properties of the observer (“what the observer expects”) to define the observed object. Hence, two independent observers might evaluate the same system differently.

The temptation to rely on more formal concepts, such as Kolmogorov complexity [7], also falls short: Because the output of the CAS comes from the CAS itself, then the size of the CAS program serves as an upper bound on the Kolmogorov complexity of the output.

Consider a simpler system, that of “Rule 110” as discussed in [11]. This is a deterministic cellular automaton that produces two-dimensional patterns that are difficult for anyone to predict. The entire CA can be expressed as a single decimal number, 110, and yet the results of running this CA are surely richer than what three decimal digits express, and in much the same way in which full-color images of the Mandelbrot set are much more interesting and complicated than than “ $z \rightarrow \lambda z(1 - z)$ for complex λ and z ” [8].

Inspection of individual agents' valuations of attribute relevance in the CAS does not reveal any strong resemblance to the global estimate, yet that global estimate is still attracted to the same profile as was produced without the CAS. There is not enough evidence to suggest that this attraction is the result of anything more complicated than summarization, but the general model of exchange-based simulations leaves open the possibility for more complicated patterns to emerge.

4 Conclusions and Future Work

The key behaviors that the authors have observed and described here are these:

- Complex adaptive simulations employing a free-market paradigm produce estimates of attribute relevance that – despite having access only to a small portion of the data – are more similar to the full non-CAS runs than are non-CAS runs when the data are sparse.
- Entropy-based information gain and similarity-based information gain report comparable estimates of attribute relevance in a non-CAS analysis using the full data set.
- Uniformly missing data more severely impact a targeted method (such as the entropy-based gain) than they impact the agnostic method (such as the similarity-based gain).

There exist good methods for answering specific questions of clean data sets. Market simulations using a CAS approach appear to represent a useful class of methods for exploring larger, less well conditioned data sets.

The elegance and danger behind CAS simulations, however, are one and the same: The amplification of simple behaviors over many independent agents can produce trends not immediately obvious from an analogous, non-CAS implementation. A sensitive dependence on initial conditions may not only make possible significantly novel solutions, but may stymie minor variations on successful methods.

The CAS simulation described above is more robust than more rigorous statistical methods such as factor analysis or principal component analysis, since, like many other machine-learning methods, it is more tolerant of missing or highly correlated data. The CAS system estimates the relevance of features in a manner comparable to C4.5 and other information-gain-based methods, but does so by acting on only very small fractions of data at a time by exploiting local interactions, and in an exchange model that is much more directly understandable by end users.

CAS methods that employ free-market simulations have proved to be an interesting and useful complement to traditional analyses. Modeling tasks such as estimating attribute relevance within free-market exchanges of simple agents has good promise, primarily because it can produce comparable results in a framework that allows for more natural extension and modification of the method: Not all end-users are mathematicians who can manually adjust information gain, but

many are businesspeople who can artfully apply what they know about buying and selling to improving the quality or efficiency of the results. Refining this model is where the authors' next work is directed.

References

1. Arteconi and D. Hales. Greedy cheating liars and the fools who believe them. Technical Report UBLCS-2005-21, University of Bologna, Dept. of Computer Science, Bologna, Italy, Dec. 2005. Also available at: <http://www.cs.unibo.it/pub/TR/UBLCS/2005/2005-21.pdf>.
2. Dooley, K. (1997): "A Complex Adaptive Systems Model of Organization Change," *Nonlinear Dynamics, Psychology, & Life Science*, Vol. 1, No. 1, p. 69-97.
3. Dorigo M., Nolfi S., Trianni V. (2006) Cooperative Hole-Avoidance in a Swarm-bot. *Robotics & Autonomous Systems*
4. Hadzikadic, M., Bohren, B.F. (1997). Learning to Predict: INC2.5. In *IEEE Trans. Knowl. Data Eng.* 9(1): 168-173.
5. Hair, Jr., J., Anderson, R., Tatham, R., Black, W. (1998) *Multivariate Data Analysis*, Fifth Edition. Prentice Hall. Upper Saddle River, NJ.
6. Holland, J.H. (1995). *Hidden Order: How Adaptation Builds Complexity*. Basic Books, New York.
7. Grunwald, P.D. and Vitanyi, P.M.B. Kolmogorov complexity and information theory. With an interpretation in terms of questions and answers. *J. Logic, Language, and Information*, 12:4(2003), 497–529.
8. Mandelbrot, B. Fractal aspects of the iteration of $z \rightarrow \lambda z(1 - z)$ for complex λ and z . *Non-Linear Dynamics* (New York, 1979). Edited by Robert H. G. Helleman. *Annals of the New York Academy of Sciences*: 357, 249-259.
9. Mushroom records drawn from The Audubon Society Field Guide to North American Mushrooms (1981). G. H. Lincoff (Pres.), New York: Alfred A. Knopf. Donated to the UCI Machine Learning repository by Jeff Schlimmer (Jeffrey.Schlimmer@a.gp.cs.cmu.edu) Date: 27 April 1987. URL: <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/mushroom/>
10. Quinlan, J.R. (1993). *C4.5 Programs for Machine Learning*. Morgan Kaufmann.
11. Weisstein, Eric W. "Rule 110." From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/Rule110.html>

Exploring Phrase-Based Classification of Judicial Documents for Criminal Charges in Chinese

Chao-Lin Liu and Chwen-Dar Hsieh

Department of Computer Science, National Chengchi University, Taiwan
chaolin@nccu.edu.tw

Abstract. Phrases provide a better foundation for indexing and retrieving documents than individual words. Constituents of phrases make other component words in the phrase less ambiguous than when the words appear separately. Intuitively, classifiers that employ phrases for indexing should perform better than those that use words. Although pioneers have explored the possibility of indexing English documents decades ago, there are relatively fewer similar attempts for Chinese documents, partially because segmenting Chinese text into words correctly is not easy already. We build a domain dependent word list with the help of Chien's PAT tree-based method and HowNet, and use the resulting word list for defining relevant phrases for classifying Chinese judicial documents. Experimental results indicate that using phrases for indexing indeed allows us to classify judicial documents that are closely similar to each other. With a relatively more efficient algorithm, our classifier offers better performances than those reported in related works.

1 Introduction

We investigate the effectiveness of applying phrases for indexing judicial documents in Chinese. Conventional wisdom and experimental results suggest that phrases provide better indications of contents of the indexed documents than keywords, thereby offering better chances of higher quality of information retrieval. Indeed, many natural languages contain homonyms and polysemes, so using isolated keywords for indexing takes the risk of interpreting words as unintended senses, and using phrases helps to alleviate the ambiguity problems with the contextual information provided by the surrounding words. Due to this intuition, Salton, Yang and Yu have pioneered the applications of phrases for indexing English documents as early as 30 years ago [1], and many researchers have followed this line of work [2, 3].

Chinese text consists of Chinese characters, and a number of consecutive characters form a word in the sense of English words. For instance, XUN (凶) and QI (器) are two Chinese characters, and XUN-QI (凶器) is a Chinese word approximately corresponding to *weapons* in English. SH-YONG-XUN-QI (使用凶器) is a Chinese phrase that contains two words, where SH-YONG (使用) means *use* in English, and SH-YONG-XUN-QI means *use weapons* in English.

Partially due to our ignorance, we have not been able to identify sufficient work that is directly related to indexing Chinese documents with phrases. More commonly,

people segment Chinese text with the help of a machine readable lexicon, and then index the documents with Chinese words. With special techniques for obtaining information about Chinese words such as Chien's PAT tree-based approach [4], one may segment Chinese text without using lexicons. Instead of going through Chinese word segmentation first, some have used character level bigrams for indexing Chinese documents [5, 6]. This approach offers a much improved performance than character-level indexing for Chinese text, while requiring a much larger space of index terms [7]. To further improve the quality of search results, some consider short Chinese words for indexing [7].

As an exploration toward phrase-based indexing of Chinese text, we consider word-level bigrams for indexing indictment documents in Chinese. To this end, we rely on both HowNet [8] and Chien's PAT tree-based methods for identifying useful Chinese words. After obtaining definitions of Chinese words, we segment each document for obtaining pairs of words, and use them as the signatures of the documents. We define the similarity between indictment documents based on the number of common term pairs. Having built this infrastructure, we classify indictment documents based on their prosecution categories as Liu did in [9], and classify indictment documents based on their cited articles as Liu did in [10]. Current experimental results indicate that using term pairs leads to classification of higher quality for the former task. However, the new method provides only comparable performance on the latter task. Our methods differ from Liu's methods for the second task in two important ways. In addition to using different indexing units, i.e., single terms vs. term pairs, we use different ways to obtain weights for these indexing units. We are still looking into the second task for further improvements.

Section 2 provides more background information regarding our work. Section 3 discusses our methods of obtaining Chinese words for the legal domain from our corpus. Section 4 extends the discussion for how we obtain phrases, how we assign weights to the phrases, and how we use the phrases for comparing the similarity between indictment documents. Section 5 contains the experimental results, and Section 6 wraps up this paper with some discussions.

2 Background

We provide more background information on details of our classification tasks. We exclude information how we segment Chinese character strings into word strings [9] for page limits. We follow a standard procedure for segmenting Chinese, i.e., preferring the longer matches while using a lexicon to determine the word boundaries, that has been adopted in the literature.

We can classify indictment documents in two different levels of grain sizes. The coarser lever is based on the prosecution categories, and the more detailed level is based on the cited articles. We consider six different prosecution categories: larceny (竊盜), robbery (搶奪), robbery by threatening or disabling the victims (強盜), receiving stolen property (贓物), causing bodily harm (傷害), and intimidation (恐嚇). We use X_1 , X_2 , ..., X_6 , respectively, to denote these categories henceforth. The criminal law in Taiwan dedicates one chapter to each of these prosecution reasons, except that the two types of robberies X_2 and X_3 occupy the same category.

Once judges determine the prosecution categories of the defendant, they have to decide what articles are applicable to the defendants. Each chapter for a prosecution category contains a few articles that describe applicability and corresponding sentence of the article. Not all prosecution categories require detailed articles that subcategorize cases belonging to the prosecution category, but some prosecution categories require more detailed specifications of the prosecutable behaviors than others. In this paper, we concern ourselves with articles 266, 267, and 268 for gambling (賭博). Article 266 describes ordinary gambling cases, article 267 describes cases that involve people who make a living by gambling, and article 268 describes cases that involve people who provide locations or gather gamblers for making profits.

Applicability of these articles to a particular case depends on details of the facts cited in the indictment documents. Very simple cases violate only one of these articles, while more complex ones call for the applications of more articles. In addition, some combined applications of these articles are more normal than others in practice. Let A, B, and C denote types of cases that articles 266, 267, and 268 are applied, respectively, and a group of concatenated letters denotes a type of cases that articles denoted by each letter are applied. Based on our gathering of the published judicial documents, we observe some common types: A, C, AB, and AC. The cases of other combinations are so rare that we cannot reasonably apply and test our learning methods at this moment. Hence we will ignore those rare combinations in this paper.

Classifying indictment documents based on the cited articles is more useful than classifying documents based on the prosecution reasons, because both legal practitioners and ordinary people benefit from more exact classification. However, classifying documents based on cited articles is distinctly more difficult than classifying documents based on prosecution reasons. Documents of lawsuits that belong to the same prosecution category contain similar descriptions of the criminal actions, and subcategorizing them requires professional training even for human experts.

3 Lexical Acquisition

Although employing a machine-readable lexicon is essential for our work, relying completely on HowNet will not provide satisfactory results. HowNet was developed by excellent researchers in China, and it is not deniable that HowNet provides invaluable information about Chinese words. Nevertheless, it is also true that the Chinese languages used in Taiwan and in China have become a bit different due to the separation in the past half century. In addition, HowNet may not include all legal terms that we need. For these reasons, we employ HowNet to find useful words for the legal applications, and Figure 1 shows the flow of how we acquire the lexical information.

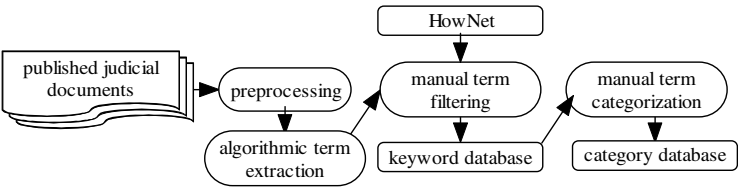


Fig. 1. Constructing databases of lexical information

We apply Chien's PAT tree-based algorithm [4] and our own algorithm, TermSpotter, for spotting possible Chinese words from a training corpus. When using the PAT tree-based algorithm, we extract only words that consist of two or three characters. We manually filter the candidate words reported by these algorithms, keeping all spotted words that were already listed in HowNet and useful words even if they are not listed in HowNet. The words are then manually clustered into categories based on their semantic similarity.

Procedure: TermSpotter (input: a training corpus; output: a list of candidate words)

1. Scan the corpus, and obtain frequencies of all bigrams
2. Concatenate bigrams that have similar occurrence frequencies into longer words, preferring those have higher frequencies
3. Save all n-grams that exceed the threshold for occurrence frequency into a word-list
4. Remove selected words from the wordlist, and return the resulting wordlist

Our method for spotting terms in our training data is actually very simple. The TermSpotter aggregates consecutive n-grams that have similar and high occurrence frequencies into a longer word. At step 2, two neighbor n-grams will be aggregated if their frequencies did not differ more than 50% of their individual frequencies. At step 3, n-grams are considered frequent if they occurred more than 30 times. The choices of 50% and 30 were arbitrary, which make TermSpotter perform satisfactorily so far. Step 4 removes words that meet specific conditions, and we subjectively set up the conditions.

We employed both TermSpotter and Chien's algorithm to look for useful terms from 10372 real world indictment documents. The very first step in processing the legal documents that were published as HTML files was to extract the relevant sections at the preprocessing step. We then ran TermSpotter and Chien's algorithm over the corpus to get the candidate words, and manually filtered the list to obtain the keyword database. At the manual filtering step, all extracted words that were also included in HowNet would be saved in the keyword database. We subjectively decided whether to save the extracted words that were not included in HowNet. After checking each of the algorithmically extracted words, we found 1847 useful words that were already included in HowNet. We also found 832 useful words for the legal application, but they were not included in the original HowNet. In total, we have 2679 words in the keyword database.

To enhance the information encoded by the words, we manually categorized words which have similar meanings in legal applications. For instance, we have a category for *location* which includes Chinese words for *banks*, *post offices*, *night markets*, etc., and we also have a category for *vehicle* which includes such Chinese words as *passenger cars*, *busses*, *taxis*, and *trucks*. We have 143 categories that include more than two Chinese words, and we treat a word as a single-word category if that word carries a unique meaning. In categorizing the words, we ignored the problem of ambiguous words, so a word was assumed to belong to only one category. Although making such a strong assumption is subject to criticism, we consider it worthy of exploration because words might carry specific meanings in phrases in legal documents.

4 Phrase-Based *k*NN Classification

Instance-based learning [11] is a technique that relies on past recorded experience to classify future problem instances, and *k*NN methods are very common among different incarnations of the concept of instance-based learning. By defining a distance measure between the past experience and the future problem instance, a system selects *k* past experiences that are most similar to the future problem instance, and classifies the future instance based on the classes of the selected *k* past experiences.

The appropriateness of the similarity measure is crucial to the success of a *k*NN-based system. Given a segmented Chinese text as we explained in Section 2, we can treat each past experience as a vector or a bag of Chinese words. The distance measure can be defined in appropriate ways [12]. In this paper, we report our experience in using phrases as units of indexing for legal documents in Chinese, and in learning the weights for phrases in classifying documents. Figure 2 shows the flow for training and testing our classifiers, and major components are explained in this section.

4.1 Identifying Phrases and Learning Their Weights

Although it is intuitive that using phrases will lead to more precise indexing of the past documents, it is far less clear about how we choose phrases from sentences [1, 2, 3]. Moreover, it is not even clear that how we define “sentences” in Chinese. Although modern Chinese writing adopts punctuations such as commas and periods, a sequence of words ended with periods do not necessarily correspond to just one sentence as they normally do in English. It is very common that a sequence of Chinese words ended with a period can be translated into multiple English sentences. In this work, we choose to use commas and periods as terminators for Chinese sentences.

Given a Chinese sentence, we can segment the sequence of characters into a sequence of words. Now, with this sequence of words, can we determine a phrase that actually stands for the main idea of the original sentence? It seems that this is a tough question that we cannot solve without resorting to semantic analysis of the original text. If we may solve this problem now, we might have found a solution to the problem of exact indexing of documents which is an essential challenge for information retrieval. To circumvent this difficulty, we record all possible word pairs formed by words in the sentence, and disregard those pairs which do not occur more than 10 times in the training documents. We then take the union of word pairs for all

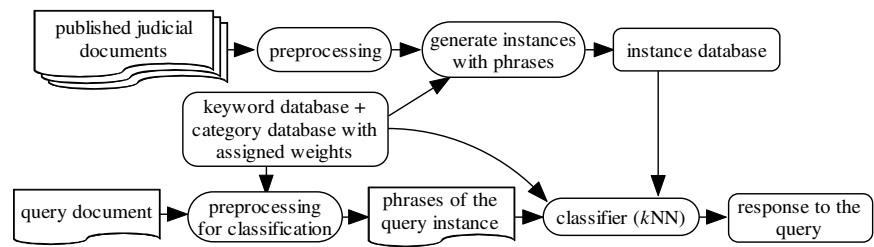


Fig. 2. Training and testing our classifier

sentences in a preprocessed document as the feature list of the document. We employ this procedure in ovals labeled with *generate instances with phrases* and *preprocessing for classification* in Figure 2.

Before we explain ways of assigning weights to phrases, we elaborate on how we define phrases in more details. Assume that, after being segmented, a sentence includes three words, α , β , γ . We will preserve the original ordering of these words, and come up with three combinations, i.e., $\alpha\text{--}\beta$, $\alpha\text{--}\gamma$, and $\beta\text{--}\gamma$, as the phrases for the sentence. As a result, if we have a document that includes this sentence, these word pairs will *all* be included in the instance that represents the original document. We recognize that this might not be a good design decision, but doing so relieves us of the task of determining which phrase is the “most representative” of the original sentence for the current exploration.

It is expected that with appropriate weights, weighted k NN methods provide better performance than plain k NN methods [11]. Hence we would also like to assign weights to phrases. The weights for phrases should reflect their potential for helping us to correctly classify documents, so defining weights based on the concept similar to the inverse document frequency [12] is desirable. As we mentioned in Section 3, we actually have converted some words to their semantic categories. Hence, we will assign weights to phrases at the level of semantic category, rather than to the phrases at the word level.

We explore two methods for assigning weights to phrases. Let $S=\{s_1, \dots, s_i, \dots, s_n\}$ be the set of different types of documents in an application. Assume that a phrase κ appears f_i times in documents of type s_i . Let p_i be the conditional probability of the current document belonging to s_i , given the occurrence of κ . We may assign the quantity defined in (1) as the weight of κ . Notice that the denominator in (1) assimilates the formula of entropy. Hence a phrase with larger w_1 will collocate with fewer types of documents. We also explore the applicability of (2). Qualitatively, w_2 is similar to w_1 in that a phrase with larger w_2 will collocate with fewer types of documents.

$$w_1(\kappa) = \frac{1}{-\sum_{t=1}^n p_t \log p_t}, \quad \text{where } p_i = \frac{f_i}{\sum_{r=1}^n f_r} \quad (1)$$

$$w_2(\kappa) = \left(\sum_{t=1}^n p_t^2 \right)^2, \quad \text{where } p_i = \frac{f_i}{\sum_{r=1}^n f_r} \quad (2)$$

4.2 Similarity Measure

Now that we have converted the original documents into instances that are represented by sets of phrases and that we have assigned weights to phrases, we are ready to define the similarity measure between instances for our classifier that adopts the k NN approach. Assume that we have two instances i_1 and i_2 , each representing a set of key phrases. Let $u_{1,2}$ denote the intersection of i_1 and i_2 . We explore two methods, shown in (3) and (4), for computing the similarity between i_1 and i_2 . The basic element in both (3) and (4) is the portion of common phrases in the phrases of the instances being compared. Two instances are relatively more similar if they share more common phrases. Formulas (3) and (4) differ only in how we combine the two ratios.

$$s_1(i_1,i_2) = (\frac{\text{total weights of } u_{1,2}}{\text{total weights of } i_1} + \frac{\text{total weights of } u_{1,2}}{\text{total weights of } i_2})/2 \tag{3}$$

$$s_2(i_1,i_2) = \frac{\text{total weights of } u_{1,2}}{\sqrt{(\text{total weights of } i_1) \times (\text{total weights of } i_2)}} \tag{4}$$

4.3 More Design Factors

In addition to how we obtain basic words, how we define weights, and how we define similarity measure between instances, there are other design decisions that we can manipulate. It is interesting to consider whether we should take into account the part of speech (POS) of the words when we construct phrases. Since our phrases consist of only two words, it is natural for us to consider only verbs and nouns in forming the phrases. Under this constraint, we could have phrases of the form verb-verb, verb-noun, noun-verb, and noun-noun. If we interpret our phrases as basic events in the descriptions of the criminal violations, we might prefer to employ phrases led only by verbs in computing similarity between instances. Hence, we can compare the performance of classification when we consider any word pairs and when we consider only phrases with leading verbs.

The other design factor that we have considered is to limit the source of phrases. Recall the way we construct phrases in Section 4.1. If a sentence contains many basic words, we would create many combinations of these basic words, potentially introducing noisy phrases into our databases. Though the noise thus resulted may not interfere the classification very much, it may degrade the computational efficiency of the classifier. This observation leads us to screen sentences from where we would obtain phrases. For the experiment results reported in the following section, we consider sentences that contain no more than 16 Chinese characters which can be segmented into no more than 3 words. Experimental results under other settings are included in a longer version of this paper.

5 Experimental Evaluations

We evaluated our classifier with real world judicial documents. Table 1 shows the quantities of the documents used for the task of classifying

Table 1. Number of cases in different prosecution categories

	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆
training	1600	1600	1600	1600	710	241
test	400	400	400	400	179	60

documents based on the prosecution categories. Table 2 shows quantities of documents for the task of classifying documents based on the cited articles. We acquired the documents from the web site of the Judicial Yuan, Taiwan (www.judicial.gov.tw). We continue to use the notation for representing different types of documents, which are discussed in Section 2.

Since our work is not different from traditional research in text classification, we embraced such standard measures as *precision*, *recall*, the *F* measure, and *accuracy*

for evaluation [12]. Let p_i and r_i be the *precision* and *recall* of an experiment, F is defined as $(2 \times p_i \times r_i) / (p_i + r_i)$. Due to page limits, we must summarize the classification quality for all different types of documents, and we took the arithmetic average of the *precision*, *recall*, F , and *accuracy* of all experiments under consideration.

Tables 3 and 4 show statistics about the performance of our classifier. These tables employ the same format. The top row indicates whether we considered all types of word pairs or only phrases that were led by verbs, as we discussed in Section 4.3. The second row indicates whether we employed formula (1) or (2) for defining weights of phrases, and the third row indicates whether we computed similarity between instances by formula (3) or (4). The numbers in the top row indicate the quantities of phrases that were obtained from the training documents.

Table 2. Number of cases in different combinations of cited articles for gambling cases

	A	C	AB	AC
training	1066	802	809	344
test	267	201	203	86

Table 3. Classification based on prosecution categories (3,16)

POS	any phrases (1504)				phrases led by verbs (989)			
Weights	(1)		(2)		(1)		(2)	
Similarity	(3)	(4)	(3)	(4)	(3)	(4)	(3)	(4)
<i>precision</i>	74.7%	81.9%	79.7%	85.7%	72.2%	77.1%	77.9%	85.5%
<i>Recall</i>	74.3%	67.3%	79.1%	81.5%	70.3%	63.4%	76.2%	81.5%
F	70.0%	68.8%	77.7%	82.1%	66.1%	63.9%	74.9%	82.2%
<i>accuracy</i>	74.9%	73.2%	81.6%	86.2%	70.7%	69.0%	78.6%	85.4%

Table 3 shows the statistics for the experiments for prosecution category-based classification, when we extracted phases from sentences which contained no more then 16 Chinese characters which were segmented into no more than three words. Using this setup, we obtained 1504 phrases when we considered all types of phrases, and, if we ignored phrases that were not led by verbs, we obtained 989 phrases from the training documents. We observed that no matter whether we consider POS of constituents of the phrases, the combination of formulas (2) and (4) would offer the best performance. This proposition held when we repeated the same experiment

Table 4. Classification based on cited articles (3,16)

POS	any phrases (459)				phrases led by verbs (262)			
Weights	(1)		(2)		(1)		(2)	
Similarity	(3)	(4)	(3)	(4)	(3)	(4)	(3)	(4)
<i>precision</i>	77.8%	73.9%	79.5%	80.6%	75.2%	74.7%	77.6%	77.9%
<i>Recall</i>	79.0%	75.4%	80.5%	81.7%	76.0%	76.1%	78.7%	79.0%
F	77.4%	73.8%	78.8%	80.2%	75.0%	74.8%	77.3%	77.7%
<i>accuracy</i>	78.9%	75.4%	80.8%	81.9%	77.0%	76.1%	79.3%	79.7%

procedure for setups where we obtained phrases from sentences of different number of characters and words. Results for classifying cases based on cited articles, i.e., statistics in Table 4, further support that (2) and (4) together outperform for the task of cited article-based classification than other combinations of the formulas. Assuming that we use (1) for defining weights, (3) seems to be a better choice for computing similarity between instances.

Statistics, particularly those for the combination of (2) and (4), in both Tables 3 and 4 do not show any relative superiority in classification quality for whether we should consider POS of the constituents of the phrases. We do not consider the differences in the statistics significant although the averages for not considering POSs seem a bit better. However, we would have used more than 40% of number of phrases in the classification for considering all types of phrases. Using phrases that were led by verbs clearly had an edge on computational efficiency.

Corresponding numbers for the combination of (2) and (4) in Tables 3 and 4 support the intuition that classification based on prosecution categories is relatively easier than classification based on cited articles. The same observation had been reported by Liu and Liao [10]. However, we have to interpret this indication of our statistics carefully, because all cases that were used for obtaining Table 4 committed the crime of gambling in different details, while cases for obtaining Table 3 belonged to a range of different prosecution categories not including gambling, as we reported in Section 2. In addition, corresponding numbers for other columns in Tables 3 and 4 do not fully support the intuition. Cases that belong to prosecution categories in our experiments may contain related criminal violations that disoriented our classifier. For instance, it should not be surprising that one would describe something that related to larceny ($\mathbf{X}_1$) before one could describe how one received stolen property ($\mathbf{X}_4$). Therefore differentiating $\mathbf{X}_1$ and $\mathbf{X}_4$ may not be easier than telling A and AC apart for gambling cases.

6 Concluding Remarks

We reported an exploration among a myriad of possible ways of applying phrases to classifying judicial documents. The preliminary results are encouraging. Compared with the results reported in [9], we are able to achieve better quality of classification when our target prosecution categories are much closer than those used in previous studies. For the task of classifying cases based on cited articles, our method provides comparable quality with those reported by Liu and Liao [10]. However, our training method is more succinct and easy to understand and implement than the introspective learning method used in [10]. Our current system employs only relative frequencies of phrases among different types of documents, but Liu and Liao had to learn and adjust weights for each keyword in each training instance. Nevertheless, our advantage may have come from that we have to manually filter the Chinese words, and Liu and Liao's approach does not need human intervention at all. Similar to Liu's results, our results are better than those reported by Thompson [11]. Our results are also better than those achieved by pioneers of the phrase-based approach [2, 3], partially because the legal domains employ more specific terms in the judicial documents.

Besides the inspiring results, the exploration left us more questions to study. Indeed, pioneers had reported many challenges for the phrase-based approach [2, 3].

It should be clear that coming up with an effective method for weighting the phrases is not easy. It is also difficult to determine how we obtain important phrases for indexing the case instances. Our two types of phrases correspond somewhat to statistical phrases and syntactic phrases [3]. Our results concur with Fagan's in that syntactic phrases do not provide significant better performance than statistical phrases. Nevertheless, we cannot be satisfied with the current results, and would like to study related issues for fully understanding the applicability of phrase-based indexing for specific domains such as legal case classification in Chinese.

Acknowledgements

This research was funded in part by contract NSC-94-2213-E-004-008 of the National Science Council of Taiwan. We thank the reviewers for their unreserved and invaluable comments. Unfortunately, we cannot add more contents for responding to the comments when we have to shorten the submitted paper for page limits already. A more complete version of this paper is available upon request.

References

ICAIL stands for *Int. Conf. on Artificial Intelligence and Law*, and SIGIR for *Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*.

1. G. Salton, C. S. Yang, and C. T. Yu, A theory of term importance in automatic text analysis, *J. of the American Society for Information Science*, **26**(1), 33–44, 1975.
2. J. L. Fagan, Automatic phrase indexing for document retrieval, *Proc. of the 10th SIGIR*, 91–101, 1987.
3. W. B. Croft, H. R. Turtle, and D. D. Lewis, The use of phrases and structured queries in information retrieval, *Proc. of the 14th SIGIR*, 32–45, 1991.
4. L.-F. Chien, PAT-tree-based keyword extraction for Chinese information retrieval, *Proc. of the 20th SIGIR*, 50–58, 1997.
5. I. Moulinier, H. Molina-Salgado, and P. Jackson, Thomson Legal and Regulatory at NTCIR-3: Japanese, Chinese and English retrieval experiments, *Proc. of the 3rd NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering*, 2002.
6. L.-F. Chien, Fast and quasi-natural language search for gigabytes of Chinese texts, *Proc. of the 18th SIGIR*, 112–120, 1995.
7. K. L. Kwok, Comparing representations in Chinese information retrieval, *Proc. of the 20th SIGIR*, 34–41, 1997.
8. HowNet. <www.keenage.com>
9. C.-L. Liu, C.-T. Chang, and J.-H. Ho, Case instance generation and refinement for case-based criminal summary judgments in Chinese. *J. of Information Science and Engineering*, **20**(4), 783–800, 2004.
10. C.-L. Liu and T.-M. Liao, Classifying criminal charges in Chinese for Web-based legal services, *Proc. of the 7th Asia Pacific Web Conf.*, 64–75, 2005.
11. T. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
12. C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.
13. P. Thompson, Automatic categorization of case law, *Proc. of the 8th ICAIL*, 70–77, 2001.

Regularization for Unsupervised Classification on Taxonomies

Diego Sona, Sriharsha Veeramachaneni, Nicola Polettini, and Paolo Avesani

ITC-IRST, Via Sommarive, 18 - 38050 - Povo - Trento - Italy
{sona, sriharsha, polettini, avesani}@itc.it

Abstract. We study unsupervised classification of text documents into a taxonomy of concepts annotated by only a few keywords. Our central claim is that the structure of the taxonomy encapsulates background knowledge that can be exploited to improve classification accuracy. Under our *hierarchical Dirichlet* generative model for the document corpus, we show that the unsupervised classification algorithm provides robust estimates of the classification parameters by performing *regularization*, and that our algorithm can be interpreted as a regularized *EM* algorithm. We also propose a technique for the automatic choice of the regularization parameter. In addition we propose a regularization scheme for *K-means* for hierarchies. We experimentally demonstrate that both our regularized clustering algorithms achieve a higher classification accuracy over simple models like minimum distance, *Naïve Bayes*, *EM* and *K-means*.

1 Introduction

With the proliferation of information on the web as well as within organizations, there is increased necessity of automatic clustering and classification of information over concept taxonomies for its organization and management.

A (concept) *taxonomy* is a hierarchy of classes created by an editor, where each class is represented by a node annotated by a set of linguistic terms (called *keywords*) and the topology (i.e., the directed edges between nodes) represents the (often implicit) relationships between classes. Each document in the corpus belongs to one class (not just the leaf classes) that best describes the object.

There is considerable previous work on hierarchical classification of textual documents in a supervised setting (e.g., *Neural Networks* [6], *SVMs* [3,9], *Probabilistic models* [2,5], *Association Rules* [8], etc.). Supervised learning, however, is not always feasible because labeled training examples are needed whenever a taxonomy is created or modified. The solution therefore is to automatically organize a given collection of unlabeled documents according to the given taxonomy.

Since the size of the taxonomy usually grows with the amount of data in the corpus, the number of documents per class is never large enough for accurate parameter estimation. The effects of sparse data on parameter estimation can be alleviated by *regularization* or *smoothing*, which are essentially variance reduction techniques. This is the motivation for the *EM* with *shrinkage* algorithm proposed by McCallum *et al.* [5]. They propose a regularization scheme based

on a generative model for classifying documents only on the leafs of a taxonomy. Under the generative model they derive an algorithm to obtain *shrinkage* estimators for class-conditional word probabilities. Self-Organizing Maps (*SOMs*) [4] are another way to perform regularized clustering.

Since the concept taxonomies are organized such that the children of a node provide more specialized information than their parent, the parameters of a node are closer to its parent and offsprings than to other nodes in the taxonomy. Our *hierarchical Dirichlet* model (presented in [7]) for the documents in the corpus incorporates this knowledge explicitly. We derived an unsupervised classification method based on the *EM* algorithm to cluster a set of text documents into a given taxonomy. As opposed to several parameters per leaf class as in [5], our model involves just one smoothing parameter which can be learnt automatically. We apply the same regularization idea to *K-means* clustering algorithm by drawing on the philosophy of *SOMs*. Our main contributions are the development of a generative model-based regularized *EM* document clustering algorithm and a regularized *K-means* algorithm. We show empirically that our algorithms achieve a significant improvement in classification accuracy over a minimum distance (referred to as *baseline*), *Naïve Bayes*, *EM* and *K-means* algorithms.

2 Hierarchical Dirichlet Model

In our generative model a document $\mathbf{d}$ is a sequence of words drawn from a vocabulary $\mathcal{V}$ with $|\mathcal{V}| = k$. If the documents are coded as vectors with a *bag-of-words* representation, a multinomial distribution is a natural choice, hence, the probability of the document $\mathbf{d}$ given the class c in the hierarchy is given by $p(\mathbf{d}|c) \propto \prod_{j=1}^k (\theta_{cj})^{d_j}$, where d_j is the number of occurrences of the j^{th} word of the vocabulary in document $\mathbf{d}$, and the parameter vector $\boldsymbol{\theta}_c = (\theta_{c1}, \dots, \theta_{ck})^T$ describes the parameters for a multinomial distribution (i.e., θ_{cj} is the probability of the j^{th} word of the vocabulary for class c).

The above equations describe the distribution of documents for a particular class, we now describe how the parameters for the classes are themselves interdependent. For the *bag-of-words* framework we can assume the parameter vectors $\boldsymbol{\theta}_c$ themselves have Dirichlet prior distributions given by $\boldsymbol{\theta}_c \sim \text{Dir}(1, \dots, 1)$ if c is the root class, and $\boldsymbol{\theta}_c | \boldsymbol{\theta}_{\text{pa}(c)} \sim \text{Dir}(\sigma \boldsymbol{\theta}_{\text{pa}(c)})$ otherwise. Here $\text{pa}(c)$ identifies the parent of class c and σ is a constant smoothing or regularization parameter.

Since the Beta distribution is a one-dimensional special case of the Dirichlet distribution and is a conjugate prior of the binomial distribution the above model can also be applied to the *bag-of-words* framework as shown in [7].

Note that in our model the parameters along a path from the root to a leaf are generated by a random walk where the parameter σ controls the step size. A large value for σ indicates that the parameter vector for a class is *tightly* bound to that of its parent. We use the uniform prior for the root node.

To learn the model we need to estimate the parameters for the class conditional distributions. For any class, its parameter vector must be estimated given the data at the class, the parameter estimate at its parent (except, of course,

for the root for which we use the uniform distribution as the prior) and the estimates of the parameters at its children (except the leaves for which we have only the parent and the data). Hence, the estimate for the parameter vector θ_c for a class c is updated iteratively given the data at the class and the previous estimates of the parameters at its neighbors. Our goal, however, is to classify a set of unlabeled documents into the classes which are labeled only by a small set of keywords. We address this problem by clustering the documents using the well-known *EM* algorithm for multinomial mixtures.

The resulting algorithm proceeds by first computing the membership of every document to each class using the parameter estimates from the previous iteration (the E-step). As we have shown in [7], the update equation for the M-step is

$$\theta_c = \frac{\sigma \theta_{\text{pa}(c)} + (\sigma + 1) \sum_{i \in \text{ch}(c)} \theta_i + \sum_{\mathbf{d}} \mathbf{d} p(c|\mathbf{d})}{\sigma + (\sigma + 1) m_c + \sum_{\mathbf{d}} p(c|\mathbf{d})}. \quad (1)$$

where m_c is the number of children for the class c , and $\text{ch}(c)$ represents the children of class c . The derivation of the algorithm is equivalent for both multinomial and binomial distributions and it is presented in [7]. We can view this EM-step as estimating the parameter vector at a class considering the parameters of its children also as part of the data under the prior coming from the parent.

From Equation (1) we note that the estimate of the parameter vector of a particular class is just a regularized maximum-likelihood estimate from the data and the corresponding estimates from the parent and children of the class. We can therefore interpret our learning algorithm as a regularized *EM* algorithm where the regularization depends on the structure of the taxonomy.

2.1 Learning σ

We take an empirical Bayesian approach, i.e., to automatically set the smoothing parameter σ by learning it from the data. When a labeled training set is available the smoothing parameter σ can be chosen by cross-validation. Since we are, however, interested in unsupervised classification, we approach the problem of learning σ by applying the principle of maximum likelihood on held-out (unlabeled) data as was done in [5]. Furthermore, to obtain maximum utility from the dataset we perform this with a leave-one-out strategy.

In other words, at each iteration of the *EM* algorithm, after the E-step (i.e., after computing the membership of every document to each class using the current parameter estimates), we calculate the likelihood $L(\sigma)$ as the product over all documents of the probability of seeing the document given that the parameter vectors were smoothed with the particular σ . The probabilities are calculated using the parameter vectors recomputed by holding-out the particular document. This can be done quite efficiently because the original parameter vectors are just weighted averages of the documents. More formally, $L(\sigma) = \prod_{\mathbf{d} \in \mathcal{D}} p(\mathbf{d} | \theta_1^{(\mathbf{d})}, \dots, \theta_C^{(\mathbf{d})})$, where $\mathcal{D}$ is the set of documents, C is the number of classes, and $\theta_j^{(\mathbf{d})}$ is computed as for Equation (1) but avoiding the “left-out” document $\mathbf{d}$ from the data set $\mathcal{D}$. In particular, the algorithm first computes the maximum likelihood estimates removing the held-out document

$$\hat{\theta}_c = \frac{\sum_{d' \in \mathcal{D} \setminus d} d' p(c|d')}{\sum_{d' \in \mathcal{D} \setminus d} p(c|d')}. \quad (2)$$

and then the parameters $\theta_c^{(d)}$ are smoothed by using the above held-out estimates in the update Equation 1. Since $L(\sigma)$ is a function of one parameter, we can use a standard maximization procedure (e.g. line search) to efficiently find the best value of σ , which is then used to perform the smoothing at the M-step.

3 Regularized K-Means

A simple alternative to the *EM* algorithm is the traditional *K-means* algorithm adopting the cosine similarity for document clustering. In order to verify how interdependencies between classes influence the learning model, we extend *K-means* by regularization drawing on the philosophy of self-organizing maps [4]. From the above discussion, classes that are closer in the hierarchy tend to have documents that are similar, hence, we enforce this during the clustering phase by a smoothing procedure that is carried out on the reference vectors by

$$\theta_c^s = \frac{\sum_{i \in \mathcal{C}} h_{c,i} \cdot \theta_i}{\sum_{i \in \mathcal{C}} h_{c,i}} \quad (3)$$

where $\mathcal{C}$ is the set of classes in the taxonomy, θ_i is the reference vector computed as average of documents in the class, and $h_{c,i}$ is a neighborhood function monotonically decreasing for increasing topological distance between the two classes c and i in the taxonomy (a Gaussian function in our implementation).

Contrary to *hierarchical Dirichlet*, we do not have any learning algorithm for finding the best smoothing parameter (σ) in the Gaussian propagation, hence, we performed the experiments with different values for σ .

Even if the above regularization is based on *SOMs*, there are some differences from the *SOM* learning algorithm. First, the smoothing parameter $h_{c,i}$ is constant, second, the regularization does not consider the weight of the classes.

4 Experimental Results and Discussion

We tested our regularized *K-means* and *EM* algorithms (*hierarchical Dirichlet*) against the unregularized versions. We could not compare our algorithms with other state-of-the-art methods like SVMs because they are primarily for supervised classification. We evaluated the approach on eight benchmark datasets (taxonomies) extracted from two well known Web directories – GoogleTM and LookSmartTM. These eight datasets correspond to different sub-directories (concept hierarchies) selected to represent a variety in the dimensionality. All the linguistic descriptions, i.e., keywords for nodes, and content of the documents (short descriptions of the linked sites) are made by the editors of the directories. The data were preprocessed by stop-words removal, stemming, and feature selection. See [1] for a detailed description of the datasets used.

Table 1. Micro- $F1$ measures for all models on various taxonomies after convergence

	Keyword Matching		Clustering		Regularized Clustering	
	Baseline	Naïve Bayes	K -means	EM	regularized K -means (best sigma)	HD
Google						
Archaeology	26.65	25.63	23.94	29.68	24.92 (0.7)	25.66
Language	21.32	24.67	19.39	30.84	20.98 (0.7)	27.41
Neuro Disorders	39.05	37.78	32.93	39.69	39.50 (0.7)	41.02
News Media	30.70	31.66	34.78	37.68	39.71 (0.7)	38.80
Looksmart						
Archaeology	16.99	18.54	19.85	25.26	27.59 (1.0)	30.21
Common Lang.	9.93	11.25	10.92	17.44	24.14 (1.0)	28.71
Movies	27.77	26.82	29.53	33.39	39.65 (0.7)	42.70
Peripherals	9.08	10.96	8.91	14.84	15.90 (1.0)	18.09

On average there are between ten and twenty documents per class and each class is labeled by less than two keywords. Moreover, the documents (i.e., the preprocessed summaries) are usually with very few words. Hence, due to the scarcity of repeating words in the preprocessed summaries, documents were encoded as *set-of-words*, i.e., binary representation.

For each experiment our EM and K -means classification algorithms were terminated when the number of documents classified differently from the previous iteration was less than 1% or after 10 iterations, which ever occurs first. We observed that best results are obtained after very few iterations. We rejected a document when there was more than one class with the highest probability. There were no rejects for any of the iterative algorithms. The performance of the algorithms is measured by the standard *micro-F1* information retrieval metric.

4.1 Results

Table 1 summarizes the classification performance ($F1$ measure) over the eight different taxonomies for various classifiers. The *baseline* classifier is just a minimum distance classifier that assigns documents to the most similar class reference vector, constructed from the keywords at the class assigning $\theta_{cj} = 1$ if word j occurs as a keyword for node c and $\theta_{cj} = 0$ otherwise. The word probabilities used by the standard *Naïve Bayes* classifier are also obtained from the keywords at the nodes just as for the initialization step of our EM-based algorithm. So, *baseline* and *Naïve Bayes* classify using only the concept keywords. The results are quite similar and no recommendation as to which is the best can be made. The performance is dataset dependent and rejection is usually very high.

The K -means and EM algorithms using the keywords as starting seeds are able to moderately improve the performance indicating that clustering the documents is a promising approach to classification. This is because the similarity in the content of the documents can be useful independently of whether they contain the appropriate keywords.

The most interesting result, which corroborates our starting hypothesis, is that the exploitation of the relational structure among the classes for regularization, on average, leads to a further improvement in accuracy. The quality of regularized K -means and, in particular, of *hierarchical Dirichlet* considerably

increases in comparison to the unregularized version. However, as previously mentioned, since, for regularized K -means, we do not have a principled way of setting the smoothing parameter σ , we performed the experiments with various values for σ , and only show the results for the best choice of σ .

5 Conclusions

Our main aim was to provide evidence that relational structure between classes in a taxonomy encapsulates background knowledge, which can be exploited for organizing a document corpus. In particular, we developed a generative model-based regularized EM clustering algorithm (*hierarchical Dirichlet*) and a SOM -based regularized K -means algorithm, both of which exploit the structure of the taxonomy to improve classification accuracy. This is due to the increasing robustness of estimators when there are too few examples. We also provided a learning algorithm for the regularization parameter for *hierarchical Dirichlet*. Finally, we provided experimental results as evidence that regularization exploiting the structure of the taxonomy can significantly improve the classification accuracy.

References

1. P. Avesani, C. Girardi, N. Polettini, and D. Sona. TaxE: a testbed for hierarchical document classifiers. Technical Report T04-04-02, ITC-IRST, 2004. <http://sra.itc.it>.
2. M. Ceci and D. Malerba. Hierarchical classification of html documents with Web-ClassII. In *Proc. of the 25th European Conf. on Information Retrieval (ECIR'03)*, volume 2633 of *Lecture Notes in Computer Science*, pages 57–72, 2003.
3. T. Hofmann, L. Cai, and M. Ciaramita. Learning with taxonomies: Classifying documents and words. In *NIPS Workshop on Syntax, Semantics, and Statistics*, 2003.
4. T. Kohonen. *Self-Organizing Maps*, volume 30 of *Series in Information Sciences*. Springer, Berlin, 2001.
5. A. McCallum and K. Nigam. Text classification by bootstrapping with keywords, EM and shrinkage. In *ACL'99 - Workshop for Unsupervised Learning in Natural Language Processing*, 1999.
6. M.E. Ruiz and P. Srinivasan. Hierarchical text categorization using neural networks. *Information Retrieval*, 5(1):87–118, 2002.
7. S. Veeramachaneni, D. Sona, and P. Avesani. Hierarchical Dirichlet model for document classification. In *ICML'05 - Proc. of Int. Conf. on Machine Learning*, 2005.
8. K. Wang, S. Zhou, and S.C. Liew. Building hierarchical classifiers using class proximity. In *Proc. of the 25th VLDB Conference*, 1999.
9. Dell Zhang and Wee Sun Lee. Web taxonomy integration using support vector machines. In *WWW '04: Proc. of Int. Conf. on World Wide Web*, pages 472–481. ACM Press, 2004.

A Proximity Measure and a Clustering Method for Concept Extraction in an Ontology Building Perspective

Guillaume Cleuziou, Sylvie Billot, Stanislas Lew,
Lionel Martin, and Christel Vrain

Université d'Orléans, Laboratoire d'Informatique Fondamentale (LIFO),
F-45067 Orléans Cedex, France

Abstract. In this paper, we study the problem of clustering textual units in the framework of helping an expert to build a specialized ontology. This work has been achieved in the context of a French project, called BIOTIM, handling botany corpora. Building an ontology, either automatically or semi-automatically is a difficult task. We focus on one of the main steps of that process, namely structuring the textual units occurring in the texts into classes, likely to represent concepts of the domain. The approach that we propose relies on the definition of a new non-symmetrical measure for evaluating the semantic proximity between lemma, taking into account the contexts in which they occur in the documents. Moreover, we present a non-supervised classification algorithm designed for the task at hand and that kind of data. The first experiments performed on botanical data have given relevant results.

1 Introduction

The exploitation of textual data from scientific basis is an ambitious goal for researches in the field of knowledge management and acquisition. One of the first steps needed to elaborate an information system is to acquire some ontology of the field considered. In this paper we deal with the problem of building a specialized ontology by a semiautomatic approach. For this purpose, we first study the step of automatic extraction of terminological classes. These classes have to be validated by an expert as concepts of the domain, before being structured in some ontology.

The task of clustering words may be done by different ways, depending on the application, the available knowledge of the field and the possible treatments. The existing methods address the following tasks: to define a measure of proximity between words and/or to propose an efficient clustering method. There exists many ways to measure semantic proximity between words; we can organize such measures into three categories: statistical, syntactical or knowledge-based measures.

Statistical measures are often based on co-occurrence of words in texts, using the Harris' hypothesis ([2]) that expresses that two semantically close words are

often found in similar contexts. These contexts may be more accurately found by a syntactic analysis of the sentences [7]. The semantic proximity of words may also be measured using a knowledge base as for example a thesaurus or an existing ontology of the field ([9,5]).

Many clustering methods, leading to different organizations of data have already been proposed and can be used to organize a set of words into classes, given a proximity measure. Generic clustering methods (for instance *c*-means [6], agglomerative hierarchical classification [10]) are the most currently used, even though some recent approaches applicable to this task have been proposed [5,8,1]. The main scope of this paper concerns more the proximity measure than the clustering method.

In this paper, we propose a new proximity measure between words. It is based on syntactic information obtained from a specialized corpus. It is coupled with an agglomerative hierarchical clustering method, which has been modified for this kind of application.

The paper is organized as follows. Section 2 presents the context of this work and some basic notions on the domain. In Section 3, we propose a proximity measure between words and Section 4 describes the clustering algorithm that we have developed for the concept extraction task. In the last part, we propose some experimental results obtained on a French corpus about botany *La Flore du Cameroun*, and a discussion on the perspectives of this work.

2 Framework

2.1 The Project BIOTIM

This work has been realized in the context of a French project, called BIOTIM (<http://www-rocq.inria.fr/imedia/biotim/>) that aims at building generic tools for analyzing large databases composed of texts and images, in the domain of biodiversity. In this project, we are interested in building a textual ontology of the domain from a corpus about botany, written in French. This corpus is different from traditional corpus, because of the structure of the sentences - very long descriptions without verbs and with very specialized terms - which for instance makes difficult a syntactic analysis, usually organized around verbs.

Let us notice that the work presented in this paper is not specified to French. It could be applied to corpus written in English, by changing the patterns that are used at the beginning of the process.

In the following, we use the corpus entitled “La Flore du Cameroun” (in English, “The flora of Cameroun”), which is composed of 37 volumes and published by the “Herbier National Camerounais”. Each volume has been digitalized, which has lead to some OCR errors (*Optical Character Recognition* errors).

2.2 Ontologies and Syntactic Dependences

Different steps¹ have been applied to the initial corpus for extracting a set of nominal groups that express syntactic links between two nouns, or between a

¹ Segmentation, morpho-syntactic tagging, lemmatization, extraction of terminologies.

noun and an adjective². For instance, from the text *tree with deciduous leaves*, we could extract the two expressions *tree with leaf* and *deciduous leaf*. They are composed of three lemmas *tree*, *leaf* and *deciduous* ; to simplify we use in the following the term “word” to denote a “lemma”.

In the following, we denote as *context of a word* w_i an expression containing w_i in which w_i has been replaced by $\sim$ (it is then a syntactic context). For instance from the two expressions *tree with leaf* and *deciduous leaf*, we get two contexts for *leaf*, namely *tree with* $\sim$ and *deciduous* $\sim$, a context for *tree*, $\sim$ *with leaf* and a context for *deciduous*, $\sim$ *leaf*.

Graph Modelization

Starting from Harris’ hypothesis, we use the syntactic dependences between words to build a graph: its vertices are the words, an edge between two vertices expresses that their associated words share some common contexts [7]. A threshold is used in order to discard artificial dependences³. Studying the structure of the graph (connex component, clique graph, ...) can lead to some preliminary semantic categories.

In our experiments, we have used a threshold equal to 10 and we have noticed (as in [7]) the presence in the graph of a quite large connex component with many small ones, which seem relevant from a semantic point of view. Nevertheless, this is not sufficient and a deeper study of the dependences must be performed to get more information on the underlying ontology.

Numeric Modelization

In [4], several indices relying on the notion of contexts are introduced:

The index $a(.,.) : a(w_i, w_j)$ is the number of shared contexts between two words w_i and w_j .

The productivity of a word, denoted by $prod(.,)$, is the number of different contexts in which this word occurs. In a similar way, the productivity of a context $prod(c)$, is the number of different words occurring in this context.

For instance, an analysis of the word *foliole* (*leaflet*) shows that 102 different expressions contain this word ($prod(foliole) = 102$). Conversely only the words (translated) *top*, *nervation*, *margin* and *leafstalk* occur in the context *leaflet with* $\sim$ ($prod(leaflet\ with\ \sim) = 4$).

The index $prox(.,.)$ formalizes the idea that if a context is very productive, its contribution for assessing the links between two words is weaker than for a less productive one. Let $\mathcal{C}_i$ (resp. $\mathcal{C}_j$) the set of contexts of the word w_i (resp. w_j), $prox$ is defined by

$$prox(w_i, w_j) = \sum_{c \in \mathcal{C}_i \cap \mathcal{C}_j} \frac{1}{\sqrt{prod(c)}}$$

² In French, such expressions have the form *Noun-Adjective* or *Noun-(Prep.(Det.))-Noun*. Nearly 35 000 expressions have been built from the corpus, containing more than 12 000 lemmas (nouns and adjectives).

³ A dependence is artificial if it involves very few contexts, mainly due to the use of “empty words”.

The index $J(.,.)$ (which is not symmetrical) formalizes the differences that may exist between a very productive word and a less productive one:

$$J(w_i, w_j) = \frac{a(w_i, w_j)}{prod(w_i)}$$

$J(w_i, w_j)$ is as higher as w_i shares many of its contexts with w_j , and has a low productivity.

Let us notice that a numeric approach leads to a loss of information: for instance, it models only the number of contexts that are shared by two words and not the list of these contexts. We show in Section 4 that we can take into account this information during clustering process.

3 A New Proximity Measure

3.1 Motivation

The index $J(.,.)$ proposed takes into account the unbalance that exists between two words, particularly in term of the number of different contexts. The main idea of the measure we propose is to compute the proximity between two words not only on the basis of their common properties (shared contexts) but also on their differences (own contexts).

Considering the two words “pétale” (*petal*) and “fleur” (*flower*), we observe on the corpus the following values: $a(\text{fleur}, \text{pétale})=54$, $prod(\text{fleur})=284$ and $prod(\text{pétale})=196$. The relative high number of common contexts (54) leads to consider that the semantic proximity between the two words is quite high. Nevertheless, we observe an important difference on one hand between the proportion of the shared contexts between “fleur” and “pétale” among the contexts of “fleur” ($\frac{54}{284} \approx 19\%$) and on the other hand between the proportion of the shared contexts between among the contexts of “pétale” ($\frac{54}{196} \approx 27.5\%$). This phenomena can be formalized in a probabilistic point of view (see next section) by the relation $P(\text{pétale}|\text{fleur}) < P(\text{fleur}|\text{pétale})$. This models the intuitive idea that giving a context c , it is more probable that if it is a context for “pétale” it is also a context for “fleur” than the inverse. In other words when describing a “pétale” we usually refer to “fleur” whereas we can describe a “fleur” without referring to its “pétale”.

Nevertheless, considering only the proportion of shared contexts among the contexts a word appear can lead to two main traps. First, a proportion gives no guarantees about the confidence of the results: for instance, a word w_i sharing its unique context with an other word w_j has a value $J(w_i, w_j)$ equal to one, whereas a word sharing 90 of its 100 contexts with an other word has a lower value for $J(w_i, w_j)$, equal to 0.9. Furthermore, the proximity should be computed not only by using quantitative information (number of contexts) but also qualitative ones (productivity of the contexts, shared or not).

In the next section, we formalize a measure taking into account these two last points.

3.2 A Non-symmetrical Measure

The proximity measure we propose is composed of two terms: the first one is an extension of the previous J coefficient and it captures the degree of inclusion of a word into an other one with respect to their associated contexts; the other term expresses a kind of confidence on the first one.

In the same way that $prox(.,.)$ extends the index $a(.,.)$, we define $\alpha(.,.) \in [0, 1]$ that improves $J(.,.)$ by introducing a notion of quality on the contexts. Let $\mathcal{C}_i$ (resp. $\mathcal{C}_j$) be the set of contexts of the word w_i (resp. w_j), $\alpha(w_i, w_j)$ is the ratio of the number of common contexts over the number of contexts of w_i , each context being weighted according to its productivity:

$$\alpha(w_i, w_j) = \frac{\sum_{c \in \mathcal{C}_i \cap \mathcal{C}_j} \frac{1}{\sqrt{prod(c)}}}{\sum_{c \in \mathcal{C}_i} \frac{1}{\sqrt{prod(c)}}}$$

The weights associated to each context must be interpreted as follows:

- the more productive the shared contexts are, the less similar the two words are,
- the more productive the own contexts of w_i are, the more w_i is similar to w_j .

The previous interpretation models the natural idea that w_i is very similar to w_j if they have strong common properties (number and interest of the shared contexts) and w_i does not differ much from w_j (number and interest of the contexts of w_i).

As mentioned above, $\alpha(.,.)$ alone is not sufficient to measure the strength of the link between two words. For the same value of $\alpha(.,.)$, it is natural to consider that a pair of words that concerns the higher number of contexts is of higher confidence. This confidence can be expressed simply by the number of common contexts (index $a(.,.)$) or more subtly by the $prox(.,.)$ index. The final non-symmetrical (or asymmetrical) [12] proximity measure $p(.,.)$ we derive is defined by the product:

$$p(w_i, w_j) = \alpha(w_i, w_j) \cdot prox(w_i, w_j)$$

To be used in further applications (*e.g.* clustering), it is necessary to propose a symmetrical version of p which retains much as the information contained in $p(.,.)$. We define the following symmetrical index $\delta(.,.)$:

$$\delta(w_i, w_j) = \min\{p(w_i, w_j), p(w_j, w_i)\}$$

In this way we consider that two words w_i and w_j are similar if and only if w_i is similar to w_j AND w_j is similar to w_i in the sense of $p(.,.)$.

3.3 Probabilistic Framework

Like for other usual measures (Mutual Information, Dice coefficient, etc.) we can propose a probabilistic interpretation of $p(.,.)$. Let $P(w_i)$ and $P(w_i, w_j)$ denote

respectively the probabilities that a context contains the unit w_i , and both the units w_i and w_j . These probabilities are approximated on a corpus as follows:

$$P(w_i) = \frac{\text{number of contexts into which } w_i \text{ appears}}{\text{total number of contexts}}$$

$$P(w_i, w_j) = \frac{\text{number of contexts into which } w_i \text{ and } w_j \text{ appear}}{\text{total number of contexts}}$$

If we omit the weights put on the contexts, the measure $p(.,.)$ we propose in this study can be rewritten in a probabilistic framework by:

$$p(w_i, w_j) \approx \frac{P(w_i, w_j)}{P(w_i)} \cdot P(w_i, w_j) \cdot N$$

where N is the total number of contexts. We then observe that p is close to the Equivalence Index ($\frac{P(w_i, w_j)^2}{P(w_i) \cdot P(w_j)}$) but it differs mainly on two original points: p introduces a weight of interest on the context and then it does not satisfy the symmetrical property⁴, as explained in section 3.2.

4 Word Clustering

Clustering a set of objects consists in finding groups of objects so that similar objects belong to the same group and dissimilar objects to distinct groups. Many strategies were proposed for this task, for example partitioning methods (c -means), hierarchical algorithms (agglomerative or divisive), mixture models, grid based methods, and so on [3].

Most of the works that deal with clustering textual data (graphic chains, lemmas, terms, key-words, documents, etc.) are more concerned with giving a meaning to the concept of proximity than with proposing a clustering method to deal with textual data. However some studies propose original approaches in order to cluster textual objects by taking into account their specificities, such as the polysemy of a word or the multi-topic aspect of a document [8,1]. Beside these marginal and recent works, the step of clustering terms remains generally realized by traditional methods (c -means or agglomerative approaches) because they are simple and easily controlled by the users.

4.1 A Dedicated Clustering Method

The approach presented here is an adaptation of the (average-link) agglomerative hierarchical algorithm. This method proceeds by successive merges of the two nearest clusters⁵, starting with the leaves (one object by cluster) and ending

⁴ Let us notice that the first term $\frac{P(w_i, w_j)}{P(w_i)}$ matches exactly with the Inclusion Index [11].

⁵ With the average-link method, the proximity between two groups is obtained by carrying out the average of the proximities between two objects of different groups.

with the root (all the objects in the same group). This kind of approach has the advantage of keeping a trace of the development of the clusters through the hierarchical tree (or dendrogram) built. On the other hand, a recurrent problem for this method is to find the relevant clusters among the set of tree nodes. In that purpose, we have chosen to prohibit the agglomeration in a cluster when this merge leads to a “conceptually non-relevant” cluster (cf. definition 1 below). The structure thus obtained is a *partial hierarchy*, i.e. a set of (small) dendrograms.

Definition 1. *Let P be a cluster consisting of the words $\{w_1, \dots, w_n\}$ and $C_1, \dots, C_n$ the sets of associated contexts with those words, P is **conceptually non-relevant** if there is no (common) context into which all the words of P appear:*

$$\bigcap_{i=1 \dots n} C_i = \emptyset$$

The algorithm we propose can be described as follows : in a first step each word is put into a cluster with itself (leaves of the hierarchy); a characterization is associated to each cluster (the set of contexts shared by its word); at each iteration, the two nearest clusters (in the sense of the average linkage) are merged if the resulting cluster respects the constraint of relevance. From this merge a new cluster is built, its characterization is the intersection of the characterizations of the two descendants. The proximity matrix between clusters is updated. When the constraint of relevance prevents from any merge, the agglomeration is stopped and the algorithm returns: the set of clusters, the hierarchical trees and the associated characterizations. We consider only the clusters containing at least two items as potential concepts.

The addition of a constraint of relevance is essential in this algorithm since it brings the guarantee that the words of a group at least appear all in a common context. Moreover the characterization of a group will help the expert to propose a label with the associated concept.

4.2 Application to the Botanical Data

We have tested the clustering algorithm on the word extracted from the botanical corpus. Among the 12 000 word, we have selected those sharing at least three contexts with another word; this restricts to 2 024 the amount of data to process. In the experiments, the proximity measure δ is used.

Figures 1 and 2 present some groups we have obtained (we present the 10 dendrograms that contain the merges that match with the 10 first iterations of the algorithm). These hierarchical trees are representative of the results. We can organize them into three categories according to the kind of words: domain-specific terms, generic terms and finally those concerning abbreviations, proper nouns or foreign words.

The trees with domain-specific terms (Figures 1 and 2, trees *c*, *d*, *e*, *f* and *j*) are difficult to evaluate for readers that are not experts of the field. Nevertheless

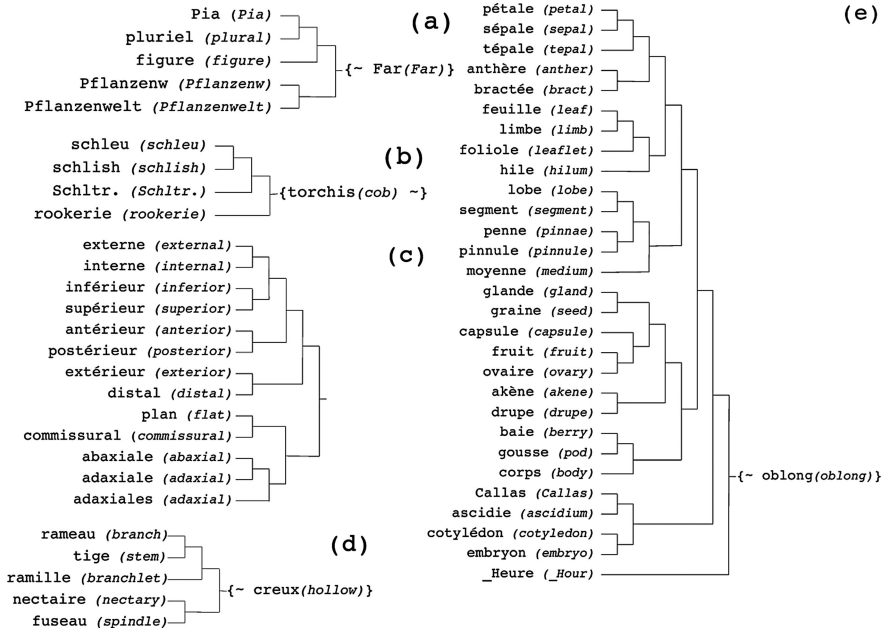


Fig. 1. Five obtained dendrograms with the characterization of the cluster in place of the root

it is possible to determine the global semantics of some clusters: for example cluster *f* is the textual representation of the concept “aspect of the limb”⁶. Other specific concepts come from the analysis of all the results, for example the “form of a sepal”, (*linear-lanceolate*, *oval-lanceolate*, *ensiform*), the “form of a leaflet” (*deltoid*, *linear*, *oval*, *rhomboid*, *falciform*, *cuneiform*, *polymorphous*), the “appearance” that a vegetable can take (*plant*, *herb*, *liana*, *treelet*), and so on.

The trees containing generic terms are easier to evaluate (Figure 2, trees *g*, *h* and *i*). Their analysis confirms the feeling of quality since the groups observed can be associated to concepts of the field:

- *g* is a textual representation of the concept “color of bark”,
- *h* is a textual representation of the concept “variations of colors” (in particular for the colors *black*, *brown* and *yellow*),
- *i* is a textual representation of the concept “metric units” (in particular to indicate the height of the plants).

These concepts can be identified more easily, due to the characterization suggested. Among the results not presented here, we find other simple concepts such as: the usual “form” associated to an element of a plant (*beak*, *dome*, *tongue*, *tape*, *gutter*), the “cardinal points” (*north*, *south*, *west*, *West*), the “month” (*January*, *May*, *August*, *November*, *December*), etc.

⁶ *Limb* : widened part of a leave or a petal.

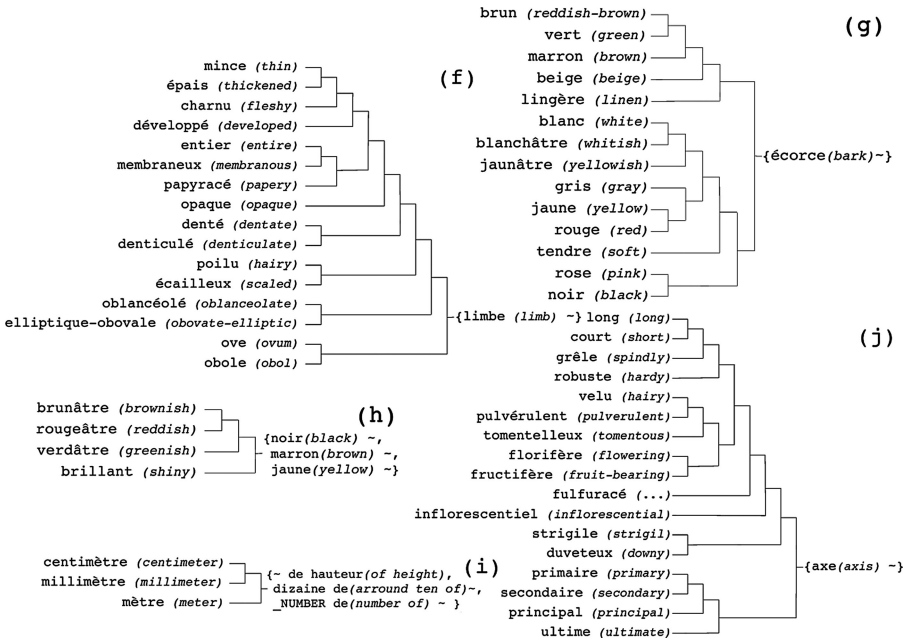


Fig. 2. Five obtained dendrograms with the characterization of the cluster in place of the root

Finally, the trees of the last category (Figure 1, trees *a* and *b*) are mainly composed of terms corresponding to abbreviations, proper nouns or foreign words. Terms of this type could be removed by selecting in the documents only the descriptive parts of plants (work under development).

To summarize, we can show the global relevance of the clusters obtained and note the helpful assistance brought by the characterization associated to each cluster. These results are explained by the measure of proximity suggested and by the adaptation of the clustering algorithm.

5 Conclusion

In this study, we focus on the task of clustering textual units in order to discover concepts, in the perspective of a semiautomatic construction of a specialized ontology. For this purpose we have first proposed a new proximity measure and then a suitable clustering method.

Unlike usual similarity measure based on the analysis of syntactic dependencies, the measure we have defined introduces weights on the contexts according to their interest; furthermore it takes into account not only the properties shared by the two textual units but also the quantity and quality of the properties that differ. The clustering method used is an adaptation of a well known hierarchical agglomerative algorithm. It produces small clusters of textual units, which share at least one common context.

In the scope of the project BIOTIM, this work has been applied on a french corpus specialized in the domain of botany. A shallow analysis first shows relevant

pairs of words, which underlines the interest of the measure proposed. Then, the classes of words discovered by the clustering process match with natural concepts of the domain.

The perspectives of this study are numerous. We are first working on an extension of the clustering algorithm leading to more flexible hierarchies. In particular, we are interested in weak hierarchies, 2-3 hierarchies and pyramids in order to allow a textual unit to appear into several classes. An other work in progress concerns the validation step that is known to be hard in this field of research. We plan to develop an interactive application allowing an expert to step in the concept formation. The interest of such an application is twofold: first to structure concepts and then to evaluate on a quantitative way the global process.

References

1. G. Cleuziou, L. Martin, and C. Vrain. PoBOC: an Overlapping Clustering Algorithm. Application to Rule-Based Classification and Textual Data. In R. López de Mántaras and L. Saitta, IOS Press, editor, *Proceedings of the 16th European Conference on Artificial Intelligence*, pages 440–444, Valencia, Spain, August 22-27 2004.
2. Z. Harris, M. Gottfried, T. Ryckman, P. Mattick, A. Daladier, T. N. Harris, and S. Harris. *The form of Information in Science: Analysis of an immunology sublanguage*. Dordrecht : Kluwer Academic Publishers, 1989.
3. A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.
4. S. Le Moigno, J. Charlet, D. Bourigault, P. Degoulet, and M.C. Jaulent. Terminology extraction from text to build an ontology in surgical intensive care. In *Proceedings of the AMIA Annual Symposium*, pages 9–13, San Antonio, Texas, 2002.
5. K. I. Lin and R. Kondadadi. A Word-Based Soft Clustering Algorithm for Documents. In *Proceedings of 16th International Conference on Computers and Their Applications*, Seattle, Washington, 2001.
6. J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical statistics and probability*, volume 1, pages 281–297, Berkeley, 1967. University of California Press.
7. A. Nazarenko, P. Zweigenbaum, J. Bouaud, and B. Habert. Corpus-based identification and refinement of semantic classes. In Daniel R. Marys, editor, *Proceedings of the 1997 American Medical Informatics Association (AMIA) Annual Fall Symposium*, Nashville, Tennessee, 1997.
8. P. Pantel. Clustering by Committee. Ph.d. dissertation, Department of Computing Science, University of Alberta, 2003.
9. R. Rada and E. Bicknell. Ranking documents with a thesaurus. *Journal of the American Society for Information Science*, 40(5):304–310, 1989.
10. P. H. A. Sneath and R. R. Sokal. Numerical Taxonomy - The Principles and Practice of Numerical Classification. San Francisco, W. H. Freeman and Compagny, 1973.
11. W. Turner, G. Chartron, F. Laville, and B. Michelet. Quantitative studies of science and technology. In A. Van Raan, editor, *Packaging Information for Peer Review: New Co-Word Analysis Techniques.*, Amsterdam: North-Holland, 1988.
12. A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.

An Intelligent Personalized Service for Conference Participants

Marco Degemmis, Pasquale Lops, and Pierpaolo Basile

Dipartimento di Informatica - Università di Bari, Bari 70126, Italy
{degemmis, lops, basilepp}@di.uniba.it

Abstract. This paper presents the integration of linguistic knowledge in learning *semantic* user profiles able to represent user interests in a more effective way with respect to classical keyword-based profiles¹. Semantic profiles are obtained by integrating a naïve Bayes approach for text categorization with a word sense disambiguation strategy based on the WordNet lexical database (Section 2). Semantic profiles are exploited by the “conference participant advisor” service in order to suggest papers to be read and talks to be attended by a conference participant. Experiments on a real dataset show the effectiveness of the service (Section 3).

1 Introduction

There are information access scenarios that cannot be solved through straightforward matching of queries and documents represented by keywords. For example, a researcher interested in retrieving “interesting papers” cannot easily express this information need as a query suitable for search engines. A possible solution could be to develop methods that discover relevant concepts from documents the user has already deemed as interesting and store them in a personal profile. Keyword-based approaches are unable to capture the *semantics* of user interests because they suffer from problems of *polysemy* and *synonymy*. Methods able to learn more accurate profiles are needed. These *semantic* profiles will contain references to concepts defined in lexicons or ontologies. Our idea was mainly inspired by the following works. *LIBRA* [8] adopts a bayesian classifier to produce content-based book recommendations by exploiting product descriptions obtained from Amazon.com Web pages. Documents are represented by using keywords and are subdivided into slots, each one corresponding to a specific section of the document (authors, title, abstract. . .). *SiteIF* [5] exploits a sense-based representation to build a user profile as a semantic network whose nodes represent senses of the words in documents requested by the user. *OntoSeek* [2]

¹ This research was partially funded by the EC under the 6th FP IST IP VIKEF No. 507173, Priority 2.3.1.7 Semantic Based Knowledge Systems (www.vikef.net) and under the DELOS NoE on Digital Libraries, Priority: Technology-enhanced Learning and Access to Cultural Heritage - Contract n. G038-507618 (www.delos.info). The work was supported also by the research project “Modelli avanzati di personalizzazione per biblioteche elettroniche”, year 2005, funded by the University of Bari.

explored the role of linguistic ontologies in knowledge-based retrieval systems. According to these works, we conceived our ITR Recommender (ITR) system as a text classifier able 1) to deal with a sense-based document representation obtained by exploiting a linguistic ontology and 2) to learn a bayesian profile from documents subdivided into slots. The strategy we propose to shift from a keyword-based document representation to a sense-based one is *to integrate lexical knowledge in the indexing step of training documents*. Several methods have been proposed to accomplish this task. Scott and Matwin proposed to include WordNet [7] information at the feature level by expanding each word in the training set with *all* the synonyms for it, including those available for each sense, in order to avoid a word sense disambiguation (WSD) process [9]. This approach has shown a decrease of effectiveness in the obtained classifier, mostly due to the word ambiguity problem. This work suggests that some kind of disambiguation is required. Subsequent works [1, 11] show that embedding WSD in document classification tasks can improve classification accuracy.

2 Learning WordNet-Based User Profiles

We consider the problem of learning user profiles as a binary text categorization task [10]: Each document has to be classified as interesting or not with respect to the user preferences. The set of categories is $C = \{c_+, c_-\}$, where c_+ is the positive class (user-likes) and c_- the negative one (user-dislikes). We present a method able to learn sense-based profiles by exploiting an indexing procedure based on WordNet, by extending the classical BOW model [10] to a model in which the senses (meanings) corresponding to the words in the documents are considered as features. The goal of the WSD algorithm is to associate the appropriate sense s to a word w in document d , by exploiting its *context* C (a set of words that precede and follow w). The sense s is selected from a predefined set of possibilities, usually known as *sense inventory*, that in our algorithm is obtained from WordNet². The basic building block for WordNet is the SYNSET (SYNONYM SET), a set of words with synonymous meanings which represents a specific sense of a word. The text in d is processed by two basic phases: The document is first tokenized and, after removing stopwords, part of speech (POS) ambiguities are solved for each token. Reduction to lemmas is performed and then synset identification with WSD is performed: w is disambiguated by determining the degree of *semantic similarity* among candidate synsets for w and those of each word in C . The proper synset assigned to w is that with the highest similarity with respect to its context of use. The semantic similarity measure adopted is the Leacock-Chodorow measure [4]. Similarity between synsets a and b is inversely proportional to the distance between them in the WordNet *is-a* hierarchy, measured by the number of nodes in the shortest path from a to b . The algorithm starts by defining the context C of w as the set of words in the same slot of w having the same POS as w , then it identifies both the sense inventory X_w for w and the sense inventory X_j for each word w_j in C . The sense inventory T for

² Version 1.7.1, <http://wordnet.princeton.edu>

the whole context C is given by the union of all X_j . After this step, we measure the similarity of each candidate sense $s_i \in X_w$ to that of each sense $s_h \in T$ and then the sense assigned to w is the one with the highest similarity score. Each document is mapped into a list of WordNet synsets following 3 steps: 1) each monosemous word w in a slot of d is mapped into the corresponding WordNet synset; 2) for each pair of words $\langle noun, noun \rangle$ or $\langle adjective, noun \rangle$, a search in WordNet is made to verify if at least one synset exists for the bigram $\langle w_1, w_2 \rangle$. In the positive case, the algorithm is applied on the bigram, otherwise it is applied separately on w_1 and w_2 ; in both cases all words in the slot are used as the context C of the word(s) to be disambiguated; 3) each polysemous unigram w is disambiguated by the algorithm, using all words in the slot as the context C of w . The WSD procedure is used to obtain a synset-based vector space representation that we called Bag-Of-Synsets (BOS). In this model, a synset vector corresponds to a document, instead of a word vector. Each document is represented by a set of *slots*. Each slot is a textual field corresponding to a specific feature of the document, in an attempt to take into account also the structure of documents. In our application scenario in which documents are scientific papers, we selected 3 slots to represent papers: *title*, *authors*, *abstract*. The text in each slot is represented according to the BOS model by counting separately the occurrences of a synset in the slots in which it appears. More formally, assume that we have a collection of N documents. Let m be the index of the slot, $n = 1, 2, \dots, N$, the n -th document is reduced to 3 bags of synsets, one for each slot:

$$d_n^m = \langle t_{n1}^m, t_{n2}^m, \dots, t_{nD_{nm}}^m \rangle$$

where t_{nk}^m is the k -th synset in slot s_m of document d_n and D_{nm} is the total number of synsets appearing in the m -th slot of document d_n . For all n, k and m , $t_{nk}^m \in V_m$, which is the vocabulary for the slot s_m (the set of all different synsets found in slot s_m). Document d_n is finally represented in the vector space by 3 synset-frequency vectors:

$$f_n^m = \langle w_{n1}^m, w_{n2}^m, \dots, w_{nD_{nm}}^m \rangle$$

where w_{nk}^m is the weight of the synset t_{nk}^m in the slot s_m of document d_n and can be computed in different ways: It can be simply the number of times synset t_k appears in slot s_m or a more complex TF-IDF score. Our hypothesis is that the proposed indexing procedure helps to obtain profiles able to recommend documents semantically closer to the user interests.

2.1 A Naïve Bayes Method for User Profiling

In the Naïve Bayes approach to text categorization, the learned probabilistic model is used to classify a document d_i by selecting the class with the highest probability. As a working model for the naïve Bayes classifier, we use the multinomial event model [6] to estimate the *a posteriori* probability, $P(c_j|d_i)$, of document d_i belonging to class c_j :

$$P(c_j|d_i) = P(c_j) \prod_{w \in V_{d_i}} P(t_k|c_j)^{N(d_i, t_k)} \quad (1)$$

where $N(d_i, t_k)$ is the number of times token t_k occur in d_i . V_{d_i} is the subset of the vocabulary containing the tokens that occur in d_i . Since each document is encoded as a vector of BOS, one for each slot, Equation (1) becomes:

$$P(c_j|d_i) = \frac{P(c_j)}{P(d_i)} \prod_{m=1}^{|S|} \prod_{k=1}^{|b_{im}|} P(t_k|c_j, s_m)^{n_{kim}} \quad (2)$$

where $S = \{s_1, s_2, \dots, s_{|S|}\}$ is the set of slots, b_{im} is the BOS in the slot s_m of d_i , n_{kim} is the number of occurrences of the synset t_k in b_{im} . ITR implements this approach to classify documents as interesting or not for a particular user. To compute (2), we only need to estimate $P(c_j)$ and $P(t_k|c_j, s_m)$ in the training phase. Documents used to train the system belong to a collection of scientific papers accepted to the 2002-2003-2004 editions of the International Semantic Web Conference (ISWC). Ratings on these documents, obtained from real users, were recorded on a discrete scale from 1 to 5. An instance labeled with a rating r , $r = 1$ or $r = 2$ belongs to class c_- (user-dislikes); if $r = 4$ or $r = 5$ then the instance belongs to class c_+ (user-likes); rating $r = 3$ is neutral. Each rating was normalized to obtain values ranging between 0 and 1:

$$w_+^i = \frac{r - 1}{MAX - 1}; \quad w_-^i = 1 - w_+^i \quad (3)$$

where MAX is the maximum rating that can be assigned to an instance. Weights in (3) are used for weighting occurrences of a token in a document and to estimate the probability terms from the training set TR . The prior probabilities of the classes are computed according to the following equation:

$$\hat{P}(c_j) = \frac{\sum_{i=1}^{|TR|} w_j^i + 1}{|TR| + 2} \quad (4)$$

Witten-Bell smoothing [12] has been adopted to compute $P(t_k|c_j, s_m)$, by taking into account that documents are structured into slots:

$$P(t_k|c_j, s_m) = \begin{cases} \frac{N(t_k, c_j, s_m)}{V_{c_j} + \sum_i N(t_i, c_j, s_m)} & \text{if } N(t_k, c_j, s_m) \neq 0 \\ \frac{V_{c_j}}{V_{c_j} + \sum_i N(t_i, c_j, s_m)} \frac{1}{V - V_{c_j}} & \text{if } N(t_k, c_j, s_m) = 0 \end{cases} \quad (5)$$

where $N(t_k, c_j, s_m)$ is the count of the weighted occurrences of the token t_k in the training data for class c_j in the slot s_m , V_{c_j} is the total number of unique tokens in class c_j , and V is the total number of unique tokens across all classes. $N(t_k, c_j, s_m)$ is computed as follows:

$$N(t_k, c_j, s_m) = \sum_{i=1}^{|TR|} w_j^i n_{kim} \quad (6)$$

In (6), n_{kim} is the number of occurrences of the token t_k in the slot s_m of the i^{th} instance. The sum of all $N(t_k, c_j, s_m)$ in the denominator of equation (5)

denotes the total weighted length of the slot s_m in the class c_j . The outcome of the learning process is a probabilistic model used to classify a new document in the class c_+ or c_- . The model is used to build the personal profile that includes those tokens (synsets) that turn out to be most indicative of the user's preferences, according to the value of the conditional probabilities in (5).

3 The “Conference Participant Advisor” Service

The “Conference Participant Advisor” service was designed to support users in planning the conference visit by exploiting their semantic profiles. The prototype was realized in context of the “International Semantic Web Conference 2004”. In order to access personalized recommendations, the conference participant must register to the service, browse the document repository (2002-2003 ISWC papers) and provide not less than 20 ratings on those papers. Then, system builds the participant profile and classifies ISWC 2004 papers by using the learned profile to obtain a personalized program in which recommended talks are highlighted. An experimental evaluation of semantic profiles was carried out on ISWC papers in order to estimate if the BOS version of ITR improves the performance with respect to the BOW one. Experiments were carried out on a collection of 100 papers (42 papers of ISWC 2002, 58 papers of ISWC 2003) rated by 11 real users. Papers are rated on a 5-point scale mapped linearly to the interval $[0,1]$. Tokenization, stopword elimination and stemming have been applied to obtain the BOW. Documents indexed by the BOS model have been processed by WSD procedure, obtaining a 14% feature reduction (20,016 words vs. 18,627 synsets). This is mainly due to the fact that bigrams are represented using only one synset and that synonyms are represented by the same synset. Classification effectiveness was evaluated in terms of *precision* and *recall* [10]. We also adopted the Normalized Distance-based Performance measure (NDPM) [13] to measure the distance between the ranking imposed by the user ratings and the ranking predicted by ITR according to the a-posteriori probability of the class *likes*. Values range from 0 (agreement) to 1 (disagreement). In the experiments, a paper is considered as *relevant* by a user if the rating > 3 , as *relevant* by ITR if $P(c_+|d_i) \geq 0.5$, computed as in equation (2). We executed one experiment for each user. Each experiment consisted in 1) selecting the ratings of the user; 2) splitting the selected data into a training set Tr and a test set Ts (using a 5-fold cross validation [3]); 3) using Tr for learning the corresponding user profile; 4) evaluating the predictive accuracy of the induced profile on Ts , using the aforementioned measures. We obtained an improvement both in precision (+1%) and recall (+2%) of the BOS model with respect to the BOW one, while NDPM has not been improved, even if it remains acceptable. A Wilcoxon signed ranked test ($p < 0.05$) has been performed. We considered each dataset as a single trial for the test, thus 11 trials have been executed. The test confirmed that there is a statistically significant difference in favor of the BOS model with respect to the BOW one only as regards recall, but not in precision.

4 Conclusions

We presented a system exploiting a bayesian learning method to induce *semantic* user profiles from documents represented using WordNet synsets. Our hypothesis is that replacing words with synsets in the indexing phase produces a more accurate document representation to infer more accurate profiles. This is confirmed by the experiments carried out to evaluate the effectiveness of the “Conference Participant Advisor” service and can be explained by the fact that synset-based classification allows to select documents with a high degree of semantic coherence, not guaranteed by word-based classification. As a future work, we plan to also exploit domain ontologies to realize a more powerful document indexing.

References

- [1] S. Bloedhorn and A. Hotho. Boosting for text classification with semantic features. In *Proc. of 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Mining for and from the Semantic Web Workshop*, pages 70–87, 2004.
- [2] N. Guarino, C. Masolo, and G. Vetere. Content-based access to the web. *IEEE Intelligent Systems*, 14(3):70–80, 1999.
- [3] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth Int. Joint Conf. on Artificial Intelligence*, pages 1137–1145. San Mateo, CA: Morgan Kaufmann, 1995.
- [4] C. Leacock and M. Chodorow. *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press, 1998.
- [5] B. Magnini and C. Strapparava. Improving user modelling with content-based techniques. In *Proc. 8th Int. Conf. User Modeling*, pages 74–83. Springer, 2001.
- [6] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *Proceedings of the AAAI/ICML-98 Workshop on Learning for Text Categorization*, pages 41–48. AAAI Press, 1998.
- [7] G. Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4), 1990. (Special Issue).
- [8] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the 5th ACM Conference on Digital Libraries*, pages 195–204, San Antonio, US, 2000. ACM Press, New York, US.
- [9] S. Scott and S. Matwin. Text classification using wordnet hypernyms. In *COLING-ACL Workshop on usage of WordNet in NLP Systems*, pages 45–51, 1998.
- [10] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 2002.
- [11] M. Theobald, R. Schenkel, and G. Weikum. Exploting structure, annotation, and ontological knowledge for automatic classification of xml data. In *Proceedings of International Workshop on Web and Databases*, pages 1–6, 2004.
- [12] I. Witten and T. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4), 1991.
- [13] Y. Y. Yao. Measuring retrieval effectiveness based on user preference of documents. *Journal of the American Society for Information Science*, 46(2):133–145, 1995.

Contextual Maps for Browsing Huge Document Collections

Krzysztof Ciesielski¹ and Mieczysław A. Kłopotek^{1,2}

¹ Institute of Computer Science, Polish Academy of Sciences,
ul. Ordona 21, 01-237 Warszawa, Poland

² Institute of Computer Science, University of Podlasie,
ul. Sienkiewicza 51, 80-110 Siedlce, Poland
{kciesiel, kłopotek}@ipipan.waw.pl

Abstract. The increasing number of documents returned by search engines for typical requests makes it necessary to look for new methods of representation of contents of the results, like document maps. Though visually impressive, doc maps (e.g. WebSOM) are extensively resource consuming and hard to use for huge collections.

In this paper, we present a novel approach, which does not require creation of a complex, global map-based model for the whole document collection. Instead, a hierarchy of topic-sensitive maps is created. We argue that such approach is not only much less complex in terms of processing time and memory requirement, but also leads to a robust map-based browsing of the document collection.

1 Introduction

The rapid growth in the amount of written information prompts for a means of reducing the flow of information by concentrating on major topics in the document flow, including the one on the World Wide Web. Clustering documents based on similar contents may be helpful in this context as it provides the user with meaningful classes or groups. One of the prominent approaches was the WebSOM project, producing a two-dimensional map of document collection, where spacial closeness of documents reflected their conceptual similarity. Hence cluster membership depends not only on other cluster members, but also on the inter-cluster 2D grid structure. This approach results in more intuitive clustering, but also imposes huge computational burden.

A recent study [4] demonstrated also deficiencies of various approaches to document organization (including WebSOM) under non-stationary environment conditions of growing document quantity, in terms of both stability and deficiency. A dynamic self-organizing neural model, so-called Dynamic Adaptive Self-Organising Hybrid (DASH) model, based on an adaptive hierarchical document organization, supported by human-created concept-organization hints available in terms of WordNet, has been proposed out of that study.

In this paper, we propose a novel approach to both the issue of topic drift and scalability. Section 2 explains in detail the concept of so-called contextual maps.

While being based on a hierarchical (three-level) approach, it is characterized by three distinctive features. In our opinion, WebSOM-like clustering is inefficient, because its target 2D grid is too rigid. Hence, we propose first to use growing neural gas (GNG) clustering technique, which is also a structural one, but has a more flexible structure accommodating better to non-uniform similarity constraints. The GNG is then projected onto a 2D map, which is less time consuming than direct WebSOM like map creation. In fact, any other structural clustering like aiNet (artificial immune system approach) could be used instead of GNG. The second innovation is the way we construct the hierarchy: first, we split the documents into (many) independent clusters (which we call "contexts"), then apply structural clustering within them, and in the end cluster structurally the "contexts" themselves. What we gain, is the possibility of drastic dimensionality reduction within the independent clusters (as they are more uniform topically) which accelerates the process and stabilizes it. The third innovation is the way we apply GNG technique. Instead of the classical global search, we invented a mixed global/local search especially suitable for GNG.

Out of these inventions, we gain speed. But not at the expense of final map quality. But what is more, the new techniques deal surprisingly well with non-stationarity of document flow - the issues of topic drift and the rapid growth of document collection. We demonstrate the validity of these claims in section 3. The last section contains some final comments on presented approach and future research directions.

2 Contextual Maps

In our work we use well known term vector space approach to document representation. It is a known phenomenon that text documents are not uniformly distributed over that space. Characteristics of frequency distributions of a particular term depend strongly on document location. In our approach we automatically identify groups of similar documents in a preprocessing step. We argue that after splitting documents in such groups, term frequency distributions within each group become much easier to analyze. In particular, it appears to be much easier to select significant and insignificant terms for efficient calculation of similarity measures during map formation step. Such document clusters we call *contextual* groups (or "contexts"). For each contextual group, separate maps are generated. To obtain more informative maps there is a need to balance (during initial contextual clustering) size of each cluster. The number of documents presented on a map cannot be too high due to rapidly growing computational time. On the other hand, a map should not hold only a few irrelevant documents.

Constraints on cluster size are matched by recurrent divisions and merges of fuzzy document groups, created by a Fuzzy C-Means (ISODATA) algorithm. There is an additional modification in optimized quality criterion that penalizes for imbalanced splits (in terms of cluster size).

In the first step, whole document set is split into a few (2-5) groups. Next, each of these groups is recursively divided until the number of documents inside

a group meets required criteria. So we obtain a tree of clusters. In the last phase, groups which are smaller than predefined constraint, are merged to the closest group¹. Similarity measure is defined as a single-linkage cosine angle between both clusters centroids.

Crucial phase of contextual document processing is the division of terms space (dictionary) into - possibly overlapping - subspaces. In this case it is important to calculate fuzzy membership level, which will represent importance of a particular word or phrase in different contexts (and implicitly, ambiguity of its meaning). Fuzzy within-group membership of the term m_{tG} is estimated as:

$$m_{tG} = \frac{\sum_{d \in G} (f_{td} \cdot m_{dG})}{f_G \cdot \sum_{d \in G} m_{dG}} \quad (1)$$

where f_G is the number of documents in cluster G , m_{dG} is the membership degree of document d in G , f_{td} is the number of occurrences of term t in d .

Next, vector-space representation of a document is modified to take into account document context. This representation increases weights of terms which are significant for a given contextual group and decrease weights of insignificant terms. In the extreme case, insignificant terms are ignored, what leads to the (topic-sensitive) reduction of space dimensionality. To estimate the significance of term in a given context, the following measure is applied:

$$w_{tdG} = f_{td} \cdot m_{tG} \cdot \log \left(\frac{f_G}{f_t \cdot m_{tG}} \right) \quad (2)$$

where f_{td} is the number of occurrences of term t in document d , m_{tG} is the degree of membership of term t in group G , f_G is the number of documents in group G , f_t is the number of documents containing term t .

As mentioned, instead of the rigid WebSOM like 2D grid, we use the more flexible GNG [2] model. Main idea behind our approach is to replace a single structural model by a set of independently created contextual models and to merge them together into a hierarchical model. Training data for each model is a single contextual group. Each document is viewed as a standard vector in term-document space, but we use w_{tdG} instead of the tfd measure.

Notice that in original GNG [2], like in WebSOM, the most computationally demanding part is the winner search phase. The replacement of global search with local one is not applicable because the GNG graph may not be connected. We propose a simple modification consisting in remembering winning node for more than one connected component of the GNG graph. To increase accuracy, we apply the well-known Clustering Feature Tree [7] to group similar nodes in dense clusters. Node clusters are arranged in the hierarchy and stored in a balanced search tree. Thus, finding closest (most similar) node for a document requires $O(\log_t V)$ comparisons, where V is the number of nodes and t is the tree branching factor (refer to [7] for details). Amortized tree structure maintenance cost (node insertion and removal) is also proportional to $O(\log_t V)$.

¹ To avoid formation of additional maps which would represent only a few outliers in document collection.

To represent visually similarity relation between contexts, additional "global" map is required. Such model becomes a root of contextual maps hierarchy. Main map is created in a manner similar to previously created maps, with one distinction: an example in training data is a weighted centroid of referential vectors of the corresponding "context": $x_i = \sum_{c \in M_i} (d_c \cdot v_c)$, where M_i is the set of cells in i -th contextual model, d_c is the density of the cell and v_c is its referential vector.

Main map cells and regions are labeled with keywords selected by our contextual term quality measure: $Q_{tG} = \ln(1 + f_{tG}) \cdot (1 - |EN_{tG} - 0.5|)$, where EN_{tG} denotes normalized entropy² of term frequency distribution within the group.

Learning process of the contextual model is to some extent similar to the classic, non-contextual learning. However, it should be noted that each constituent model can be processed independently, even in parallel. Also a partial incremental update of such models is better manageable in terms of model quality, stability and time complexity. The incrementality is in part a consequence of the iterative nature of the learning process. So if new documents come, we can consider the learning process as having been stopped at some stage and it is resumed now with all the documents. We claim that it is not necessary to start the learning process from scratch neither in the case that the new documents "fit" the distribution of the previous ones nor when their term distribution is significantly different. This claim is supported by experimental results presented in the section 3.2. In the section 3.3 we present some thoughts on scalability issues of contextual approach.

3 Experimental Results

To evaluate the effectiveness of the presented contextual approach, we compared it to the "from scratch" map formation. The architecture of our visual search engine BEATCA [5] supports comparative studies of clustering methods at the various stages of processing of document collection. In this paper we focus on evaluation of the GNG winner search method and the quality and stability of the resulting incremental clustering model with respect to the topic-sensitive learning approach. Below we describe the overall experimental design, quality measures used and the results obtained. The incrementality study in section 3.2 required manually labeled documents, so the experiments were performed on a subset of widely-used "20 Newgroups" document collection. The scalability study in section 3.3 was based on a collection of more than one million Internet documents, crawled by our topic-sensitive crawler.

3.1 Quality Measures for the Document Maps

A document map may be viewed as a special case of the concept of clustering. One can say that clustering is a learning process with hidden learning criterion. The criterion is intended to reflect some esthetic preferences, like: uniform split into groups (topological continuity) or appropriate split of documents with

² Entropy divided by the number of the terms in the group.

known a priori categorization. As the criterion is hidden, in the literature [8,1,3] a number of clustering quality measures have been developed, checking how the clustering fits the expectations. We selected the following ones for our study:

- **Average Map Quantization:** the average cosine distance between each pair of adjacent nodes. The goal is to measure topological continuity of the model (the lower this value is, the more "smooth" model is): $AvgMapQ = \frac{1}{|N|} \sum_{n \in N} \left(\frac{1}{|E(n)|} \sum_{m \in E(n)} c(n, m) \right)$, where N is the set of graph nodes, $E(n)$ is the set of nodes adjacent to the node n and $c(n, m)$ is the cosine distance between nodes n and m .
- **Average Document Quantization:** average distance (according to cosine measure) for the learning set between the document and the node it was classified into. The goal is to measure the quality of clustering at the level of a single node: $AvgDocQ = \frac{1}{|N|} \sum_{n \in N} \left(\frac{1}{|D(n)|} \sum_{d \in D(n)} c(d, n) \right)$, where $D(n)$ is the set of documents assigned to the node n .
- **Average Weighted Cluster Purity:** average "category purity" of a node (node weight is equal to its density, i.e. the number of assigned documents): $AvgPurity = \frac{1}{|D|} \sum_{n \in N} \max_c (|D_c(n)|)$, where D is the set of all documents in the corpus and $D_c(n)$ is the set of documents from category c assigned to the node n . Similarly, *Average Weighted Cluster Entropy* measure can be calculated, where $D_c(n)$ term is replaced with the entropy of the categories frequency distribution.
- **Normalized Mutual Information:** the quotient of the entropy with respect to the categories and clusters frequency to the square root of the product of category and cluster entropies for individual clusters [1].

All measures range from 0 to 1. First two describe smoothness of inter-cluster transitions and cluster "compactness" (the lower the better). The other two evaluate the agreement between the clustering and the a priori categorization of documents (i.e. particular newsgroup in case of newsgroups messages). The higher the value is, the better agreement between clusters and a priori categories.

3.2 Incrementality Study

Model evaluation were executed on 2054 of documents downloaded from 5 newsgroups with quite well separated main topics (antiques, computers, hockey, medicine and religion). Each GNG network has been trained for 100 iterations with the same set of learning parameters, using our new winner search method.

In the main case (depicted with the black line), network has been trained on the whole set of documents. This case was the reference one for the quality measures of adaptation as well as comparison of the winner search methods.

Figure 1 presents comparison of a standard global winner search method with our own CF-tree based local approach. Standard local search method (used in SOM) is considered since it is completely inappropriate in case of unconnected graphs. Obviously, tree-based local method is invincible in terms of computation

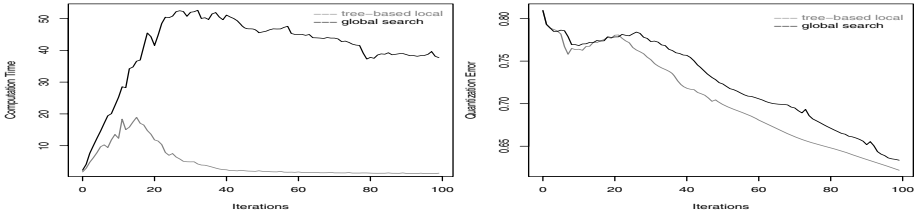


Fig. 1. Winner search methods (a) computation time (b) model quality

time. The main drawback of the global method is that it is not scalable and depends on the total number of nodes in the GNG model.

The results seemed to be surprising at first glance. Initially, the quality was similar, later on - global search appeared to be worse of the two! We have investigated it further and it turned out to be the aftermath of process divergence during the early iterations of the training process. We'll explain it later on the example of another experiment.

In the next experiment on topic drift, in addition to the main reference case, we had another two cases. During the first 30 iterations network has been trained on 700 documents only. In one of the cases (light grey line, massive document addition) documents were sampled uniformly from all five groups and in the 33rd iteration another 700 uniformly sampled were introduced to training. After the 66th iteration the model has been trained on the whole dataset.

In the last case (dark grey line, incremental document insertion with topic drift) initial 700 documents were selected only from two groups. After the 33rd iteration of training, documents from the remaining newsgroups were gradually introduced in the order of their newsgroup membership. It should be noted here that in this case we had an a priori information on the categories of documents. In the general case, we are collecting fuzzy category membership information from Bayesian Net model [5].

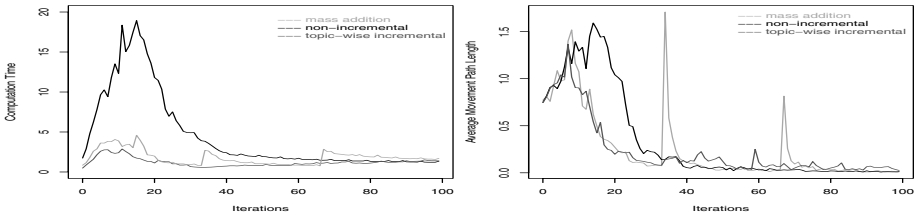
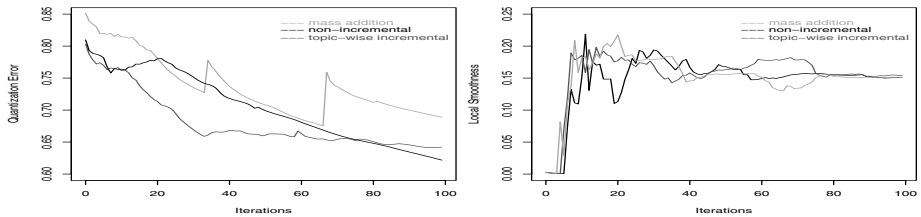


Fig. 2. Computation complexity (a) execution time of a single iteration (b) average path length of a document

As expected, in all cases GNG model adapts quite well to the topic drift. In the global and the topic-wise incremental case, the quality of the models were comparable, in terms of Average Document Quantization measure (see figure 3(a)), Average Weighted Cluster Purity, Average Cluster Entropy and Normalized Mutual Information (for the final values see table 1). Also the subjective

Table 1. Final values of model quality measures

	<i>Cluster Purity</i>	<i>Cluster Entropy</i>	<i>NMI</i>
non-incremental	0.91387	0.00116	0.60560
topic-wise incremental	0.91825	0.00111	0.61336
massive addition	0.85596	0.00186	0.55306

**Fig. 3.** Model quality (a) Average Document Quantization (b) Average Map Quantization

criteria such as visualizations of both models and the identification of topical areas on the SOM projection map were similar.

The results were noticeably worse for the massive addition of documents, even though all covered topics were present in the training from the very beginning and should have occupied their own, specialized areas in the model. However, it can be noticed on the same plot that a complex mixture of topics can pose a serious drawback, especially in the first training iterations. In the global reference case, the attempt to cover all topics at once leads learning process to a local minimum and to subsequent divergence (what, in fact, is quite time-consuming as one can notice on figure 2(a)).

As we have previously noticed, the above-mentioned difficulties apply also to the case of global winner search (figure 1(b)). The quality of the final models when we take advantage of the incremental approach is almost the same for global search and CF-tree based search (Cluster Purity: 0.92232 versus 0.91825, Normalized Mutual Information: 0.61923 versus 0.61336, Average Document Quantization: 0.64012 versus 0.64211).

The figure 2(b) presents average number of GNG graph edges traversed by a document during a single training iteration. It can be seen that a massive addition causes temporal instability of the model. Also, the above mentioned attempts to cover all topics at once in case of a global model caused much slower stabilization of the model and extremely high complexity of computations (figure 2(a)). The last reason for such slow computations is the representation of the GNG model nodes. The referential vector in such node is represented as a balanced red-black tree of term weights. If a single node tries to occupy too big portion of a document-term space, too many terms appear in such tree and it becomes less sparse and - simply - bigger. On the other hand, better separation of terms which are likely to appear in various newsgroups and increasing "crispness" of topical areas during model training leads to highly efficient computations and

better models, both in terms of previously mentioned measures and subjective human reception of the results of search queries.

The last figure, 3(b), compares the change in the value of Average Map Quantization measure, reflecting "smoothness" of the model (i.e. continuous shift between related topics). In all three cases the results are almost identical. It should be noted that extremely low initial value of the Average Map Quantization is the result of the model initialization via broad topics method [5].

3.3 Scalability Issues

To evaluate scalability of the proposed contextual approach (both in terms of space and time complexity), we built a model for a collection of more than one million documents crawled by our topic-sensitive crawler, starting from several Internet news sites (cnn, reuters, bbc). Resulting model consisted of 412 contextual maps, which means that the average density of a single map was about 2500 documents. Experimental results in this section are presented in series of box-and-whisker plots, which allows to present a distribution of a given evaluation measure (e.g. time, model smoothness or quantization error) over all 412 models, measured after each iteration of the learning process (horizontal axis). Horizontal line represents median value, area inside the box represents 25% - 75% quantiles, whiskers represent extreme values and each dot represents outlier values.

The whole cycle of map creation process took 2 days. It is impressing result, taking into account that Kohonen and his co-workers reported processing times in order of weeks [6]. It should also be noted that the model was built on a single personal computer (Pentium IV HT 3.2 GHz, 1 GB RAM). As it has been stated before, contextual model construction can be easily distributed and parallelized, what would lead to even shorter execution times.

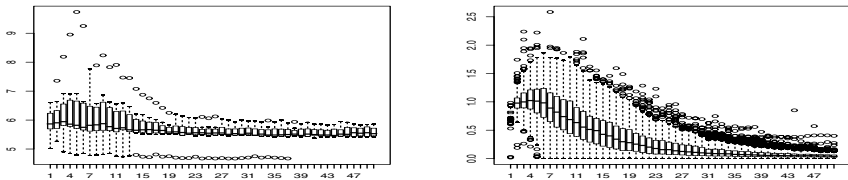


Fig. 4. Contextual model computation complexity (a) execution time of a single iteration (b) average path length of a document

The first observation is the complexity of a single iteration of GNG model learning (Figure 4(a)), which is almost constant, regardless of the increasing size of the model graph. It confirms the observations from section 3, concerning efficiency of the tree-based winner search methods. One can also observe the positive impact of homogeneity of the distribution of term frequencies in documents grouped to a single map cell. Such homogeneity is - to some extent - acquired by initial split of a document collection into contexts. Another cause of the processing time reduction is the contextual reduction of vector representation dimensionality, described in the section 2.

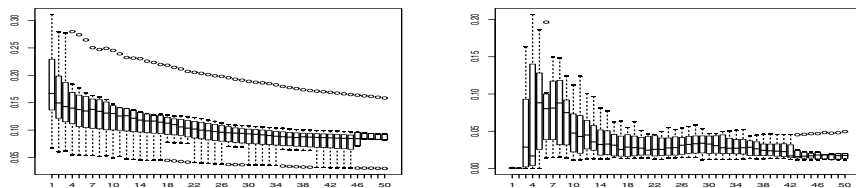


Fig. 5. Contextual model quality (a) Average Document Quantization (b) Average Map Quantization

In the Figure 4(b), the dynamic of the learning process is presented. The average path length of a document is the number of shifts over graph edges when documents is moved to a new, optimal location. It can be seen that model stabilizes quite fast; actually, most models converged to final state in less than 30 iterations. The fast convergence is mainly due to topical initialization. It should also be noted here that the proper topical initialization can be obtained for well-defined topics, which is the case in contextual maps.

The Figure 5 presents the quality of the contextual models. The final values of average document quantization (Figure 5(a)) and the map quantization (Figure 5(b)) are low, which means that the resulting maps are both "smooth" in terms of local similarity of adjacent cells and precisely represent documents grouped in a single node. Moreover, such low values have been obtained for moderate size of GNG models (majority of the models consisted of only 20-25 nodes - due to their fast convergence - and represented about 2500 documents each).

4 Concluding Remarks

As indicated e.g. in [4], most document clustering methods, including the original WebSOM, suffer from their inability to accommodate streams of new documents, especially such in which a drift, or even radical change of topic occurs.

Though one could imagine that such an accommodation could be achieved by "brute force" (learning from scratch whenever new documents arrive), but there exists a fundamental technical obstacle for a procedure like that: the processing time. The problem is deeper and has a "second bottom": the clustering methods like those of WebSOM contain elements of randomness so that even re-clustering of the same collection may lead to a radical change of the view of the documents.

From the point of view of incremental learning under SOM, a crucial factor for the processing time is the global winner search for assignment of documents to neurons. We were capable to elaborate a very effective method of mixing global with local winner search which does not deteriorate the overall quality of the final map and at the same time comes close to the speed of local search.

The experimental results indicate that the real hard task for an incremental map creation process is when documents with new topical elements are presented in large portions. But also in this case the results proved to be satisfactory.

We presented the contextual approach, which proved to be an effective solution to the problem of massive data clustering. It is mainly due to: (1) replacement of a flat, global, graph-based meta-clustering structure with a hierarchy of topic-sensitive models and (2) introduction of contextual term weighting instead of standard *tfidf* weights so that document clusters can be represented in different subspaces of a global vector space. With these improvements, we proposed a scalable approach to mining and retrieval of text data.

Contextual approach leads to many interesting research issues, such as context-dependent dictionary reduction and keywords identification, topic-sensitive document summarization, subjective model visualization based on particular user's information requirements, dynamic adaptation of the document representation and local similarity measure computation. Especially, the user-oriented, contextual data visualization can be a major step on the way to information retrieval personalization in search engines.

Clustering high dimensional data is both of practical importance and at the same time a big challenge, in particular for large collections of text documents. Still, it has to be stressed that not only textual, but also any other high dimensional data (especially characterized by attributes of heterogeneous and correlated distributions) may be clustered using the presented method.

References

1. C. Bouslis, M. Ostendorf, Combining multiple clustering systems, Proceedings of 8th European conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2004), LNAI 3202, Springer-Verlag, 2004
2. B. Fritzke, A self-organizing network that can follow non-stationary distributions, in: Proceeding of the International Conference on Artificial Neural Networks '97, Springer, 1997, pp.613-618
3. Halkidi,M., Batistakis,Y., Vazirgiannis,M.: On clustering validation techniques. Journal of Intelligent Information Systems, 17(2-3), pp.107-145, 2001
4. C. Hung, S. Wermter, A constructive and hierarchical self-organising model in a non-stationary environment, International Joint Conf. on Neural Networks, 2005
5. M. Kłopotek, S. Wierzchon, K. Ciesielski, M. Draminski, D. Czerski, Conceptual maps and intelligent navigation in document space (in Polish), to appear in: Akademicka Oficyna Wydawnicza EXIT Publishing, Warszawa, 2006
6. T. Kohonen, S. Kaski, P. Somervuo, K. Lagus, M. Oja, V. Paatero, Self-organization of very large document collections, Helsinki University of Technology technical report, 2003, <http://www.cis.hut.fi/research/reports/biennial02-03>
7. T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: Efficient data clustering method for large databases, in: Proceedings of ACM SIGMOD International Conference on Data Management, 1997
8. Y. Zhao, G. Karypis, Criterion functions for document clustering: Experiments and analysis, at <http://www-users.cs.umn.edu/~karypis/publications/ir.html>

Classification of Polish Email Messages: Experiments with Various Data Representations

Jerzy Stefanowski and Marcin Zienkiewicz

Institute of Computing Science, Poznań University of Technology,
ul. Piotrowo 2, 60-965 Poznań, Poland

Jerzy.Stefanowski@cs.put.poznan.pl, marcas@op.pl

Abstract. Machine classification of Polish language emails into user-specific folders is considered. We experimentally evaluate the impact of different approaches to construct data representation of emails on the accuracy of classifiers. Our results show that language processing techniques have smaller influence than an appropriate selection of features, in particular ones coming from the email header or its attachments.

1 Introduction

An automatic categorization of emails into multiple folders could help users in filtering too many incoming emails and organizing them in a structure corresponding to different user's topics of interest. We are interested in applying machine learning methods to find the user's folder assignment rules based on the examples of his previous decisions [2,4]. Let us remark the *folder categorization* is a more general problem than an identification of spams only [1]. It is also different to traditional text categorization as email messages are poorly structured comparing to longer texts and are written in an informal way. Moreover, there is no standard way of preparing email representation and there have been no multiple folder benchmark data sets; a recent exception is the Enron corpus [3].

Because of the page limit we reduce discussion on related works, see e.g. reviews in [1,3]. Shortly speaking, the current research are focused on evaluating accuracy of various classifiers created by learning approaches with indication to Naïve Bayes, support vector machines, k-nearest neighbor or boosted decision trees. Most of these studies concerned spam filtering than folder categorization.

Unlike these research we focus our attention on issues of constructing proper data representation, i.e. transforming emails into a data format suitable for learning algorithms. The most commonly used is the *bag of words* representation. Documents are represented as vectors of feature values referring to the importance of words selected from messages. The features are mainly extracted from text parts of an email *subject* or its *body*. Some researchers use also other features acquired from the email header and from *attachment* files [2,3,4]. However, the role of these features in obtaining efficient classifiers is not enough discussed.

Moreover, nearly all research concerned English emails. There are no such works for the Polish language, which has more complex inflection and less strict

word order. Our previous study on Web page clustering showed that preprocessing of documents in Polish influenced the results more than in English [5].

Therefore, the aim of our paper is to experimentally study the influence of different ways of constructing data representation on the accuracy of typical classifiers in the task of categorization Polish language emails into user-specific folders. As there is no corpora of real email messages written in Polish we have first to collect them. In data representations, we want to explore the use of additional information available from the header and attachment parts, which were not commonly applied in the previous research. Moreover, we will examine the different ways of handling Polish language stemming and word extraction.

2 Data Sets and Their Representation

We collected three different Polish language email data sets. The first one, further called *MarCas*, is a copy of personal emails received by the second author of this paper since October 2003 till October 2004 and contained 4667 messages. Nine different categories (folders) were used within this period (the number of emails is reported in brackets): *BooBoo* – name of the student group (218 messages), *Humor* – jokes, stories (259), *pJUG* – Java group (179), *Work* (1119), *Family* (232), *Study* (587), *Friends* (289), *Spam* (1521), *Other* (263). The other set, called *DWeiss*, is coming from our close collaborator Dawid Weiss and contains the copy of his mailbox from a longer period: October 2001-2004. Unfortunately, it concerns spam detection only and contains 9725 messages from two categories legitimate *mail* – 5606 and *spam* – 4119. Both collections contained complete information about header sections and all attachments.

To extend information about multiple folders we constructed a third data set as a collection of 36260 messages from newsgroups in Polish recorded since March till April 2005. The *NewsGroups* contains messages from the following groups: *pl.comp.lang.java* (2971), *pl.comp.lang.php* (4896), *pl.rec.kuchnia* – Polish cuisine (4137), *pl.rec.muzyka.gitara* – guitar music (7844), *pl.regionalne.poznan* – regional news about the city of Poznan (4896), *pl.regionalne.poznan* – news about Warsaw (6604), *pl.sci.fizyka* – physics (2566), *pl.sci.metamatyka* – mathematics (2286). Unfortunately, available information on messages is restricted, i.e. no attachments were stored in the NewsGroups and header parts were somehow restricted to main sections only, e.g. a section on message routing was missing.

Let us discuss a construction of data representations. The text content of the *email body* part was the basis for extracting features for the bag of words representation. The *header* part includes the set of pairs (*parameter-value*) – their meaning is defined in RFC standards. Similar to some researchers, we parsed the file to identify the following elements: *Subject*, sender information (*From*, *Replay to*) and recipient (*To*, *CC*, *Bcc*). The subject field was handled in the same way as the body. Other elements were parsed to get the complete email addresses or nicknames. We also distinguished the case of multiple recipients, i.e. each led to a separate feature. The recipient numbers were additional numeric features. Unlike previous researchers we constructed next features based on information about

Table 1. Cardinality of features of given types in studied data

data	header	servers	subject	body	attach
MarCas	99	54	15	160	75
DWeiss	42	28	11	137	283
NewsGroups	24	0	12	198	0

other parameters stored in the header. Quite important for further analysis were *Received* parameters describing the route of the message by computer servers. We also defined a numeric attribute representing the number of servers, which were recorded for the given message. Additional features included the message encoding, tools for handling it, date, parts of the day, size of the message.

Besides these extended header features, we used more features about attachments, i.e. numeric ones representing the number of attached files of a given type (zip, jpg, exe), names of the most frequent files. Contents of non-binary attachment files were processed in the same way as the body or the subject and their terms were used in the final representation.

While processing text parts we encountered the problems of *stemming*, *word extraction* and handling extra *formatting tags* in HTML. Although previous researchers were not consistent on their role in English emails [3,2,4], the proper use of stop words and stemming is more influential on processing Polish documents [5]. To examine it in email classification we created different versions of each data set: either with or without stemming. We used a quasi-stemmer for Polish called *Stempelator* introduced by Weiss the text mining framework *Carrot*²; for more details see www.carrot2.org. Similarly we studied two versions of word extraction: (1) with white characters tokenization only, (2) also with handling the additional characters as “@”, “\$” or “!”. As to email bodies written in the HTML format we also tested two techniques: either these tags were removed or not – their names were used as additional parameters. To sum up, depending on the above choices we had 8 versions of the data sets.

Because of page limits we skip details of all carried out experiments with creating classifiers from these data sets and present conclusions only. For all data sets tokenization should be extended by extra characters. Stemming was particularly useful for DWeiss data while being not so significant for others. HTML tags should be removed from MarCas and NewsGroups and should stay in DWeiss data (perhaps because of their specific use in spam).

As the number of features in data representations was quite high (around few thousands) we decided to reduce their number. Firstly, while creating bag of words representation, we removed too frequent and too infrequent terms. Then, we employed a typical feature selection mechanism based on the following measures: *information gain*, *gain ratio* and χ^2 statistics. The single features were ranked according to these measures. For each data set, we selected the features occurring in the first positions over mean values of the evaluation measures in these rankings. The general characteristics of reduced data is given in Table 1, where we distinguished the following types of features depending on their origin:

body, subject, servers, attachments, header (additional features created from other parts of the header, in particular senders).

3 Experiments

Three learning algorithms were chosen to construct email classifiers: k-nearest neighbor (k-NN), decision tree and Naïve Bayes. For running our experiments we used the Weka toolkit¹. J48 (Weka C4.5 decision tree implementation) was used with standard options. IBk – a k-NN algorithm implementation was tested with the number of neighbors equal to 1, 3 and 5 and weighted Euclidean distances. The best results were obtained for $k = 3$. In Naïve Bayes we used an option of discretizing numerical attributes.

Performance of classifiers was evaluated with the following measures: *total accuracy*, *recall*, *precision* and *F-measure*. In our experiments we used *F-measure* with the aggregation function which assigned equal importance to both precision and recall. All these measures were calculated for each category/ folder. Stratified 10-fold cross validation was applied for all data sets and we report averaged results. They are presented in Tables 2 and 3. Because of page limits, Table 3 contains results for 3-NN (IB3) only, as it performed similarly to J48 and they both were better than Naïve Bayes.

Table 2. Total accuracy (%) for different versions of data representations

data set	learning algorithm	feature set		
		subject	body	all
MarCas	IB3	61.0	70.3	88.3
	J48	61.4	70.2	87.7
	Naïve Bayes	59.2	48.3	80.0
DWeiss	IB3	82.9	83.5	98.6
	J48	82.7	82.6	98.4
	Naïve Bayes	82.3	79.3	82.2
NewsGroups	IB3	32.7	57.3	59.3
	J48	32.6	66.4	70.6
	Naïve Bayes	30.0	51.9	66.7

4 Discussion of Results and Conclusions

Let us discuss results of experiments. Comparing learning algorithms we noticed that Naïve Bayes produced significantly worse classification accuracy than other algorithms for all versions of MarCas and DWeiss data sets – the accuracy was at least 10% lower. For NewsGroups the difference of accuracy was not so large. Looking for the best classifiers, we could say that for e-mail data MarCas and DWeiss the performance of decision tree and kNN algorithms was comparable,

¹ see www.cs.waikato.ac.nz/ml/weka

Table 3. Performance of classifiers for separate folders. For each measure, three successive numbers [in %] refer to three types of features sets: subject, body and all features.

data set	folder	Recall			Precision			F-measure		
MarCas	BooBoo	84.4	70.9	86.9	97.2	97.2	97.2	88.9	69.9	87.9
	Humor	13.7	9.9	70.8	60.4	44.2	71.4	22.1	15.8	71.1
	pJUG	93.4	57.2	96.4	95.8	96.2	97.2	94.3	72.2	96.6
	Work	82.2	73.5	90.3	41.8	75.5	90.1	55.4	72.8	91.3
	Family	29.2	72.8	91.8	72.2	90.6	93.1	41.7	83.0	92.6
	Study	35.7	53.9	83.5	87.9	67.5	85.4	50.8	59.7	85.4
	Friends	11.3	30.4	49.5	80.0	58.7	53.7	19.5	40.1	49.6
	Spam	76.5	96.2	97.6	74.0	63.2	95.6	75.2	77.3	98.1
DWeiss	Other	28.5	66.1	71.9	66.3	80.4	77.5	36.8	72.8	74.6
	mail	85.1	79.3	98.4	81.3	98.3	98.7	86.0	83.1	98.8
NewsGroups	spam	76.3	80.2	97.8	82.4	72.2	97.9	77.6	83.2	98.4
	java	9.3	65.6	66.5	44.5	67.2	64.6	10.5	51.0	44.5
	php	38.6	78.7	80.0	47.0	78.3	78.5	38.6	63.3	64.4
	kuchnia	22.6	62.2	64.8	43.6	62.4	63.7	23.7	55.0	55.3
	gitara	40.4	70.7	76.1	28.4	65.7	75.3	40.3	62.3	65.8
	poznan	31.1	68.3	72.6	43.8	70.8	73.8	31.2	60.4	64.4
	warszawa	29.4	60.8	66.0	27.8	59.3	64.8	30.1	50.8	55.7
	fizyka	31.6	57.0	63.2	51.4	61.1	67.5	31.8	54.0	54.7
	matematyka	9.1	61.6	68.4	23.4	65.0	71.4	9.4	56.6	58.1

while J48 tree algorithm was slightly better for NewsGroups. The problem of detecting spam was easier task than multiple folder categorization.

More noticeable is the choice of features while creating the data representation. In Table 1 we see the high number of finally selected features about attachments (in particular for spam data), route of emails by servers and other parts of header (e.g. senders). Such information was not available for NewsGroups, which may partly explain its lower classification results. For all data sets we observed an increase of classification accuracy when applying all these additional features. Using exclusively either the subject or the body features led to lower accuracy for all types of learned classifiers. For MarCas and NewsGroups data the subject features were the least useful while for DWeiss data they were more important (perhaps it is connected with spam properties).

These observations are confirmed by the analysis of classifiers performance in separate classes – see Table 3. It is clearly illustrated by changes of the F-measure. Although these are results for 3-NN classifier, quite similar numbers were obtained for J4.8 decision trees, while slightly worse again for Naïve Bayes. Considering other measures it seems that additional features affected recall more than precision. For some folders in MarCas data the influence of using all features is really high, e.g. see recall in categories Humor, pJug, Work, Study or Family. On the other, it had lower influence on precision, e.g. in folder BooBoo, pJuG or Study. For NewsGroups, the use of the subject features only did not contribute too much to recall and precision, while the body features had also more influence on recall than on precision. Depending on the given folder, some results for the body features were quite comparable to the use of all features. For detecting spams, i.e. DWeiss data set, the use of all features increased both precision and

recall, and its influence was stronger for the spam category than regular emails. For precision the subject's features had higher influence on detecting spam while the body features were quite informative for classifying legitimate mails.

Comparing our results to related works on folder categorization of English emails, we could notice that researchers stress mainly the role of the body parts, see e.g. [3]. Although some others say that features coming from recipient or sender email parts are also useful (e.g. [4]), there is still lack of experimental evaluations. An exception is the recent study [3], the influence of different feature subsets on SVM classifiers was studied with conclusion that combining additional features improves the F measure and the body or sender features are more suitable than ones coming from the subject part.

Finally, we have to admit that we used only 3 data sets, so we should be cautious with making general conclusions. On the other hand, according to our best knowledge this research problem has not been examined yet in the context of Polish language and there have been no ready data sets. So, our original contribution is collecting at least these three data sets having different characteristics and making experiments with evaluating usefulness of various ways of constructing data representations. Saying this we risk conclusions that language processing techniques, as e.g. stemming, had smaller influence on the final classifiers than extending feature subsets by ones coming from the header part of an email and information about its attachments.

Acknowledgment. The research was supported from grant no. 3 T11C 050 26.

References

1. R. Bekkerman, A. McCallum, G. Huang: Automatic categorization of email into folders: Benchmark Experiments on Enron and SRI Corpora. U. Mass CIIR Report IR-418, 2004.
2. J. Clark, I. Koprinska, J. Poon: Linger – a smart personal assistant for e-mail classification. Proc. of the 13th Intern. Conf. on Artificial Neural Networks (ICANN'03), 2003, 274–277.
3. B. Klimt, Y. Yang: The enron corpus: a new dataset for email classification research. In Proc. of the ECML'04 Conference, 2004, 217–226.
4. G. Manco, E. Masciari, M. Ruffolo, A. Tagarelli: Towards an Adaptive Mail Classifier. Workshop "Apprendimento Automatico: Metodi ed Applicazioni", (AIIA 02). Siena, September 10-13, 2002
5. J. Stefanowski, D. Weiss: Carrot² and language properties in Web Search Results clustering. In Proc. of the 1st Atlantic Web Intelligence Conference AWIC-2003, LNCS 2663, 2003, 401-407.

Combining Multiple Email Filters Based on Multivariate Statistical Analysis

Wenbin Li^{1,3}, Ning Zhong^{1,2}, and Chunnian Liu¹

¹ The International WIC Institute, Beijing University of Technology
Beijing 100022, P.R. China
ai@bjut.edu.cn

² Department of Information Engineering, Maebashi Institute of Technology
460-1 Kamisadori-Cho, Maebashi-City 371-0816, Japan
zhong@maebashi-it.ac.jp

³ Department of Information Engineering, Shijiazhuang University of Economics
Shijiazhuang, Hebei Province 050031, P.R. China
oh_my_sun2003@hotmail.com

Abstract. In this paper, we investigate how to combine multiple e-mail filters based on multivariate statistical analysis for providing a barrier to spam, which is stronger than a single filter alone. Three evaluation criteria are suggested for cost-sensitive filters, and their rationality is discussed. Furthermore, a principle that minimizes the error cost is described to avoid filtering an e-mail of “Legitimate” into “Spam”. Comparing with other major methods, the experimental results show that our method of combining multiple filters has preferable performance when appropriate running parameters are adopted.

Keywords: Information overload, spam, e-mail filter, Naive Bayes filter, combining multiple filters.

1 Introduction

E-mail is one of the most successful computer applications yet devised. As e-mail becomes a more prevalent salesmanship of e-business, dealing with **spam** is becoming more and more costly and time consuming. There is now a great deal of interest in e-mail clients that allow a technically naive user to easily construct a personalized client for filtering spam.

To solve the problem of spam, many e-mail clients provide a filtering function based on **keyword-spotting** rules. In a “keyword-spotting” rule, the primitive conditions test to see if a word appears (or does not appear) in a certain field of an e-mail message. However, this method has two disadvantages at least. First, constructing and maintaining rules for filtering is a burdensome task. Second, any changes to the keywords will result in restructuring the filtering rules.

These scenarios provide ample ground for automated e-mail filtering to positively affect the experience of the average e-mail users. In recent years, a growing body of researches has applied machine learning techniques to the e-mail filtering

task, including rule learning [1], Naive Bayes [2,3], memory based learning [4], neural network [5], support vector machines [6], or a combination of different learners [7]. In these approaches a classifier is learned from training data rather than constructed by hand, which results in better and more robust classifiers.

Although the filtering task seems to be a simple instance of the more general text categorization task, it shows an obvious characteristic, i.e., the spam filtering errors are not of equal importance. Individuals usually prefer conservative filters that tend to classify spam as legitimate, because missing a legitimate message is more harmful than the opposite error [8]. A cost model is imperative to avoid the risk of missing legitimate e-mails. Hence, an ideal filter is the one which never *classifies legitimate as spam* (called **reject-error** in the remainder of this paper) and hardly ever makes the *inverse error* (called **receive-error** in the remainder of this paper).

The paper focuses on implementing a cost-sensitive filter by combining multiple filters and dealing with cost-oriented evaluation. The goal of combining learned filters is to obtain a more accurate prediction than can be obtained from any single source alone. Richard Segal, et al. pointed out that "... We believe that *no one anti-spam solution is the right answer*, and that *the best approach is a multifaceted one*, combining various forms of filtering ..." [9].

This motivates us to propose an algorithm called W-Voting for effectively addressing the issues mentioned as above. W-Voting is a cost-sensitive approach that adopts correspondence analysis [10] to generate the weight for each filter and eliminate the correlation of errors of each filter. The single filter in W-Voting is a Naive Bayes (NB) based method, because the NB classifier has been found particularly attractive for the task of text categorization because it performs surprisingly well in many application areas despite its simplicity [11].

The remainder of this paper is organized as follows. In Section 2, we describe the working flow of W-Voting. We then present the core algorithm and investigate the principle of **expecting to minimize the cost of prediction** in Section 3. We suggest three evaluation criteria and show our preliminary experimental results in Section 4. Finally, we give conclusions and some directions for future work in Section 5.

2 The Overview of W-Voting

Figure 1 shows the architecture of W-Voting. The algorithm consists of two main phases: training and filtering. The first phase is used to train multiple filters. The second one is used to combine these filters to classify new e-mails. Furthermore, the training phase is divided into four steps: first, dividing the training data set into Q partitions; second, training the k^{th} NB filter on the k^{th} subset ($k = 1, 2, \dots, Q$); third, constructing a **training matrix** (includes the **spam matrix** and the **legitimate matrix**); fourth, computing weights for filters using correspondence analysis. When a new e-mail arrives, Q filters will classify it according to their own weights. The CA algorithm with respect to Figure 1 will be given in Section 3.1.

From Figure 1, we can see that the key problem in W-Voting is how to build the training matrices. To simplification, we only consider the **spam matrix**. Suppose $TM_{(N \times (Q+1))}$ is the matrix, N is the total count of the training spam. The i^{th} line of TM is used to represent the i^{th} training spam, which mainly reflects the performance of each filter on the i^{th} training e-mail. Let the i^{th} line vector in TM be v_i , and $v_i = \langle p_{i,1}, p_{i,2}, \dots, p_{i,Q-1}, p_{i,Q}, p_{i,Q+1} \rangle$, where $p_{i,k} (k = 1, \dots, Q)$ is the posterior probability of the i^{th} training e-mail belonging to spam, which is computed by the k^{th} filter, and $p_{i,Q+1} = 1$.

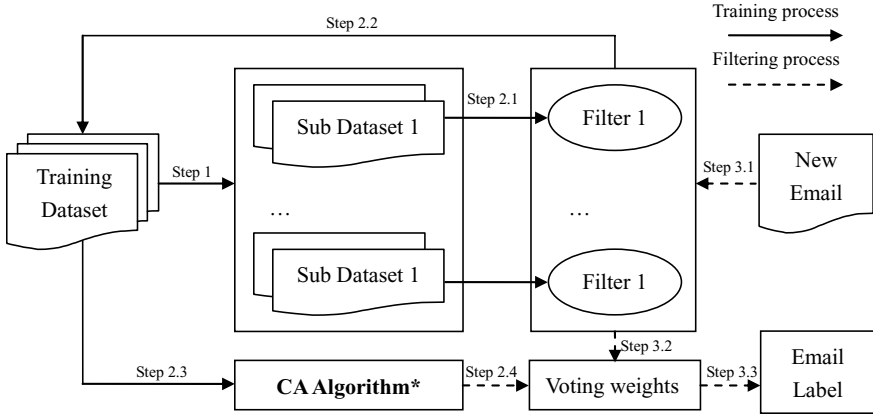


Fig. 1. The working flow of W-Voting

W-Voting is a special method of combining NB filters. A Bayesian filter assumes that an e-mail is generated by a mixture model with parameters θ , consisting of components $C = \{c_0; c_1\}$ that correspond to the classes. An e-mail is generated by first selecting a component $c_j \in C$ ($j = 0, 1$) according to the prior distribution $P(c_j|\theta)$, and then choosing an e-mail e_i according to the parameters of c_j with a distribution $P(e_i|c_j; \theta)$. The likelihood of an e-mail is given by the total probability:

$$p(e_i|\theta) = \sum_{j=0}^1 p(c_j|\theta)p(e_i|c_j; \theta). \quad (1)$$

However, the true parameters θ of the mixture model are not known. Hence, the parameters need to be estimated from the training e-mails. In this work, we adopt **Multi-variate Bernoulli Model** [12] to estimate θ and compute the probability of an e-mail given a class from the probabilities of the words given the class.

Given a new e-mail e , classification of e is performed by computing the posterior probability of each class by applying Bayes' rule:

$$p(c_j|e; \theta) = \frac{p(c_j|\theta)p(d|c_j; \theta)}{p(d|\theta)}. \quad (2)$$

The classifier simply selects the class with the highest posterior probability. Note the $p(d|\theta)$ is the same for all classes, thus e can be classified by computing:

$$c_d = \operatorname{argmax}_{c_j \in C} [p(c_j|\theta)p(e|c_j;\theta)]. \quad (3)$$

3 The W-Voting Algorithm

3.1 The CA Algorithm in W-Voting

Correspondence Analysis (CA) provides an useful tool for analyzing the association between rows and columns of a contingency table. The contingency table is a two-entries frequency table where the joint frequencies of two qualitative variables are reported. For instance, a (2×2) table could be formed by observing from a sample of n individuals with two qualitative variables: the individual's sex and whether the individual smokes. The table reports the observed joint frequencies. In general $(n \times p)$ tables can be considered.

The main idea of correspondence analysis is to develop simple indices that will show the relations between the row and column categories. These indices tell us simultaneously which column category has a bigger weight in a row category and vice-versa. Correspondence analysis is also related to the issue of reducing the dimension of the table.

We here describe the correspondence analysis process in W-Voting for the **spam matrix**. For the matrix $TM_{(N \times (Q+1))}$, the calculations of correspondence analysis may be divided into three main stages (see Algorithm 1). The first stage (Steps 1.1-1.6 in Algorithm 1) consists of some preprocessing calculations performed on $TM_{I \times J}$ ($I = N, J = Q + 1$) which leads to the standardized residual matrix. In the second stage (Step 2 in Algorithm 1), a singular value decomposition (SVD) is performed on the standardized residual matrix to redefine it in terms of three matrices: $U_{I \times K}$, $\sum_{K \times K}$, and $V_{K \times J}$, where $K = \min(I - 1, J - 1)$. In the third stage (Steps 3.1-3.2 in Algorithm 1), we use U , $\sum$, V to determine $Y_{I \times K}$ and $Z_{J \times K}$, the coordinates of the rows and columns of TM , respectively, in the new space.

$Y_{I \times K}$ is the principal coordinates of rows of TM , and $Z_{J \times K}$ is the principal coordinates of columns of TM . Note that not all K dimensions are necessary. Hence, we can reduce the dimension to $\tilde{K}$, while some information will be lost. Definition 1 is used to reflect the degree of information loss.

Definition 1. *Information Loss (IL) is defined as follows.*

$$IL = 1 - \sum_{i=1}^{\tilde{K}} \lambda_i / \sum_{i=1}^K \lambda_i \quad (4)$$

where λ_i is the diagonal element in $\sum$.

According to Definition 1, given a value of IL , the user can compute the $\tilde{K}$. In the new geometric representation, rows z_i and z_j ($i, j = 1, 2, \dots, Q$) in Z , corresponding to columns i and j in TM , will lie close to one another when

filters i and j receive similar predictions from the collection of training e-mails. Similarly, Z_{Q+1} is corresponding to the $Q+1$ column of TM , which represents the real categorization of the training e-mails. Hence, we can define:

Definition 2. The i^{th} **Filter's Weight on Spam (WJ_i)** is defined as follows.

$$WJ_i = \sum_{j=1}^{\tilde{K}} |Z_{ij} + Z_{(Q+1)j}| / \sum_{j=1}^{\tilde{K}} |Z_{ij} - Z_{(Q+1)j}| \quad (5)$$

where Z_{ij} is the j^{th} element in Z_i , $i = 1, 2, \dots, Q$.

Similarly, we can use correspondence analysis on the **legitimate matrix**. After that, we can work out the i^{th} **Filter's Weight on Legitimate (WL_i)**.

Algorithm 1. The CA algorithm in W-Voting

Data: TM . //The training matrix

Result: U, Σ, V, Z .

Process:

Step 1.1. $sum = \sum_{i=1}^I \sum_{j=1}^J tm_{i,j}$;

Step 1.2. $P = (1/sum)TM$;

Step 1.3. $r = \langle r_1, r_2, \dots, r_I \rangle$, $r_i = P_{i+}$ ($i = 1, 2, \dots, I$);

Step 1.4. $c = \langle c_1, c_2, \dots, c_J \rangle$, $c_i = P_{+i}$ ($i = 1, 2, \dots, J$);

Step 1.5. $D_r = \text{diag}(r_1, r_2, \dots, r_I)$, $D_c = \text{diag}(c_1, c_2, \dots, c_J)$;

Step 1.6. $\tilde{P} = D_r^{-1/2}(P - rc^T)D_c^{-1/2}$;

Step 2. $\tilde{P} = U\Sigma V^T$;

Step 3.1. $Y = D_r^{-1/2}U\Sigma$;

Step 3.2. $Z = D_c^{-1/2}V\Sigma$;

Step 4. return U, Σ, V, Z .

3.2 The Filtering Method

In the combining phase, the i^{th} filter's prediction or vote for spam has a strength proportional to its assigned weight WJ_i . Thus, we can compute the posteriori probability of spam as follows:

$$p(c_1/e) = \sum_{i=1}^Q WJ_i * p_i(c_1/e) \quad (6)$$

where $p_i(c_1/e)$ is the posteriori probability of spam computed by the i^{th} filter, e is a new e-mail.

Similarly, we can work out the posteriori probability of legitimate for e :

$$p(c_0/e) = \sum_{i=1}^Q WL_i * p_i(c_0/e). \quad (7)$$

In general, the decision rule selects the class for which the posteriori probability, $p(c_j/e)$ ($j = 0$ or 1), is the largest one. This way minimizes the probability of making an error. While for the e-mail filtering problem, we should consider a somewhat different rule that minimizes an expected loss or risk. Let c_{01} be the cost assigning an e-mail e to c_0 (legitimate) when $e \in c_1$ (spam), i.e., c_{01} is the cost of **receive-error**; and c_{10} is the cost of **reject-error**. In practice, it may be very difficult to assign costs. In some situations, c_{01} and c_{10} may be measured in monetary units that are quantifiable. However, in many situations, costs are a combination of several different factors measured in different units, such as money, time, quality of life. As a consequence, they may be the subjective opinion of an expert.

The overall expected cost (OEC) is used to denote the error cost of a filter:

$$OEC = c_{10}p(c_1/c_0)p_0 + c_{01}p(c_0/c_1)p_1 \quad (8)$$

where p_0 and p_1 are the prior probabilities of c_0 and c_1 , respectively, and $p(c_1/c_0)$ is the probability of making a **reject-error**, and $p(c_0/c_1)$ is the probability of making a **receive-error**.

In order to minimize OEC, we can deduce the decision rule as follows:

Theorem 1. *Given a new e-mail e , we classify it into spam if and only if:*

$$\frac{p(c_1/e)}{p(c_0/e)} \geq \frac{c_{10}p_0}{c_{01}p_1}. \quad (9)$$

Proof. Let $f_0(x)$ be the distribution function of legitimate, $f_1(x)$ be the distribution function of spam, and X_1 denotes that (as yet unknown) region in the feature space where the filter decides c_1 and likewise for X_0 and c_0 , and then write our OEC equation in terms of conditional risks:

$$OEC = c_{10}p_0 \int_{x_1} f_0(x)dx + c_{01}p_1 \int_{x_0} f_1(x)dx. \quad (10)$$

We use the fact that $\int_{X_1} f_1(x)dx + \int_{X_0} f_1(x)dx = 1$ to rewrite the risk as:

$$OEC = c_{01}p_1 + \int_{x_1} [c_{10}p_0f_0(x) - c_{01}p_1f_1(x)]dx. \quad (11)$$

We seek the minimum of OEC, so that $c_{10}p_0f_0(x) - c_{01}p_1f_1(x)$ should be lower than (or equal to) 0 in region X_1 . Hence Theorem 1 is correct.

In Theorem 1, the right hand of the inequation is a constant, we call this constant α below.

4 Evaluation Criteria and Experimental Results

4.1 Evaluation Criteria

In cost-insensitive classification tasks, two commonly used evaluation measures are **precision** and **recall**. These measures do not take the cost of two types

of errors into account. In our opinion, an ideal filter should have the following features: (1) the **reject-error rate** is 0 or in a sustainable range $(0, \beta]$, where $\beta > 0$; (2) the **receiver-error rate** is a low value; and (3) the total error cost is very low.

In order to judge whether a filter has such good features or not, we define three evaluation criteria below.

Definition 3. *The reject-error rate (RJER):*

$$RJER = F_{10}/(F_{10} + F_{00}). \quad (12)$$

Definition 4. *The receive-error rate (REER):*

$$REER = F_{01}/(F_{01} + F_{11}). \quad (13)$$

Definition 5. *The total error cost (TEC):*

$$TEC = c_{10}p_0 * RJER + c_{01}p_1 * REER. \quad (14)$$

In the three definitions, F_{10} is the count of reject-error, F_{01} is the count of receive-error, F_{00} is the times correctly classifying legitimate e-mails, and F_{11} is the times correctly classifying spam, respectively. RJER and REER reflect the ratio of two types of errors, respectively. TEC represents the total cost generated by these two types errors. A larger RJER shows that the filter makes the **reject-error** more often, and a larger REER indicates that the filter makes the **receive-error** more often. In general, the effect of RJER on TEC is more strong than REER since c_{10} is larger than c_{01} . A larger TEC indicates a worse performance.

4.2 Experimental Results

Our experiments have been performed on the Ling-Spam and PU1 corpus [13]. The PU1 corpus consists of 1099 messages, 481 of which are marked as spam and 618 are labelled as legitimate, with a spam rate of 43.77%. The messages in PU1 corpus have header fields and html tags removed, leaving only subject line and mail body text. To address privacy, each token was mapped to a unique integer. The corpus comes in four versions: with or without stemming and with or without stop word removal. The Ling-spam corpus was collected by the same author of PU1 corpus, which includes: 2412 legitimate messages from a linguistic mailing list and 481 spam messages collected by the author with a 16.63% spam rate. Like PU1 corpus, four versions are available with header fields, html tags, and attachments removed. Since the Ling-Spam corpus was compiled from different sources: the legitimate messages came from a spam-free, the topic-specific mailing list, and the spam mails were collected from a personal mailbox, the mail distribution is less like that of the normal user's mail stream, which may make messages in Ling-Spam corpus easily separable.

Table 1 shows us the distribution of testing and training e-mails of those two corpus. In our experiments, the ratio of feature subset selection is 1%; $Q = 8$;

Table 1. The distribution of testing and training e-mails of these two corpus

	the count of training e-mails		the count of testing e-mails	
	legitimate	spam	legitimate	spam
PU1	488	384	122	96
Ling-Spam	1929	384	483	97

the feature subset selection method is based on Information Gain (IG) [14]; we adopt $c_{01} = 0.5$ and $c_{10} = 2$ when we evaluate filters.

Figure 2 shows the comparative results of five filtering algorithms on those two corpus. These methods are NB, Rocchio, Vote, cost-SVM (C-SVM) and W-Voting. In this experiment, α is set to 1 for W-Voting. On both corpus, all the RJER values of W-Voting are very small. That is, W-Voting makes very few positive errors on these two corpus. Also, C-SVM makes few reject-errors on both two test data sets. However, C-SVM has more higher REER and TEC than W-Voting’s on Ling-Spam and PU1. Figure 2 (a) tells us that W-Voting has the lowest RJER, REER and TEC on PU1. On Ling-Spam, it seems that Rocchio filter is the best one. While this filer is not a cost-sensitive method on the one hand, and on the other hand, it makes more reject-errors than W-Voting on PU1. From Figure 2 (b), we can see that all filters make few positive errors on Ling-Spam. Therefore, in Ling-Spam, maybe most of legitimate test e-mails locate into the side represented by legitimate training data, and are far way from the classifying hyperplane between legitimate and spam. In other words, it is difficult for filters to classify legitimate test e-mails wrongly on Ling-Spam.

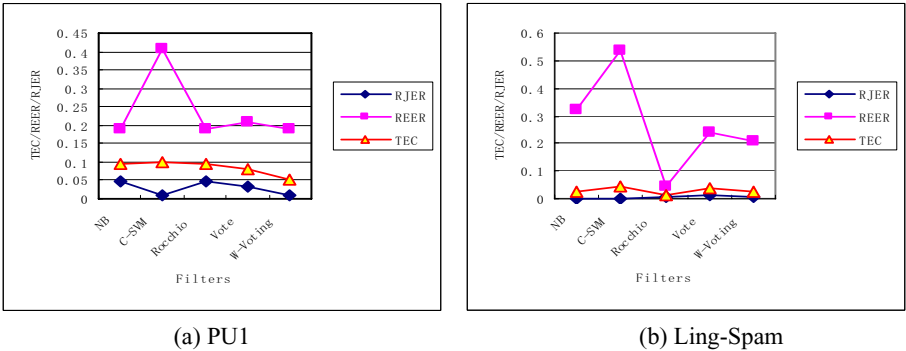


Fig. 2. The results of five filters on PU1 and Ling-Spam

Figure 3 gives the performance of W-Voting when α is changed. Figure 3 (a) and Figure 3(b) show the performance on PU1 and Ling-Spam respectively. From Figure 3 (a), we can see that RJER becomes more and more lower when α is changed to larger, while REER is opposite. When we adjust α to be greater,

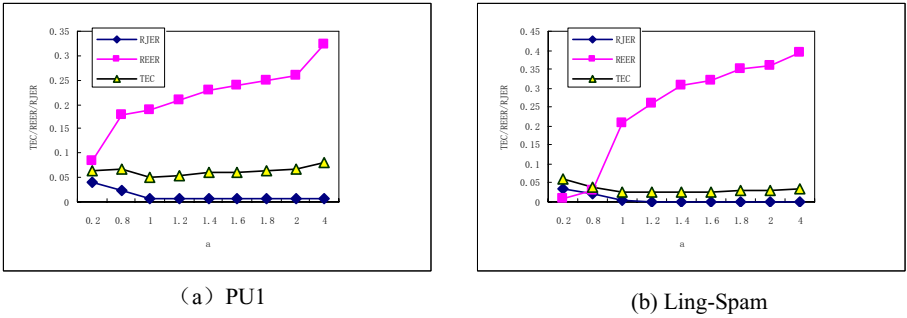


Fig. 3. The performance of W-Voting on PU1 and Ling-Spam when α is changing

TEC become more and more lower at first, when α arrives at a threshold, TEC starts to become larger adagio. Figure 3 (b) displays the same variety.

From Figure 3 (a), we know that TEC and RJER reach the lowest point when α adopts the threshold value 1. And Figure 3 (b) tells us that we should set α to the threshold value 1.2 if we want to gain the lowest TEC and RJER on Ling-Spam. In real applications, we suggest that users should adopt the value of α which is a little larger than the threshold at which TEC and RJER gain their lowest value. For example, if we use PU1 as a data set, we can set α to 1.1, and if we use Ling-Spam as a data set, we can set α to 1.3.

5 Conclusions

A voting method based on weights is presented. We use the correspondence analysis to generate the weight for each filter, then suggest two liner formulas to compute the posterior probability of spam and legitimate respectively. In order to avoid making the **reject-errors**, we deduce a principle of minimizing expected error cost. In addition, three evaluation criteria are provided to measure the cost-sensitive filters.

Experiments have been carried out on five different filters using two publicly available e-mail corpora. The main conclusion obtained from the experiments is that the W-Voting is a cost-sensitive method with ideal performance, in particular, when α is set to a proper value.

Our future work includes extending the W-Voting method to text classification with multiple classes.

Acknowledgments

The work is (partially) supported by the NSFC major research program: “Basic Theory and Core Techniques of Non-Canonical Knowledge” (NO. 60496322), Open Foundation of Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, and Project(NO. 06213558) of Dept. of Science and Technology of Hebei Province.

References

1. Elisabeth, C., Judy, K., Eric, M.: Automatic induction of rules for e-mail classification. In: Proceedings of the Australasian Document Computing Symposium, (2001).
2. Androutsopoulos, I., Koutsias, J., Chandrinou, K.V. et al.: An experimental comparison of Naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In: Proceedings of the 23rd ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece (2000) 160-167.
3. Sahami, M., Dumais, S., Heckerman, D. et al.: 1998. A bayesian approach to filtering junk e-mail. In: Proceedings the AAAI Workshop on Learning for Text Categorization, Madison Wisconsin (1998) 55-62.
4. Soonthornphisaj, N., Chaikulseriwat, K., Tang-On, P.: Anti-spam filtering: a centroid-based classification approach. In: Proceedings of the International Conference on Signal Processing, (2002) 1096-1099.
5. James, C., Irena, K., Josiah, P.: A neural network based approach to automated e-mail classification. In: Proceedings of the IEEE/WIC International Conference on Web Intelligence, (2003) 702-711.
6. Sun, D., Tran, Q.A., Duan, H. et al.: A novel method for Chinese spam detection based on one-class support vector machine. *Journal of Information and Computational Science*, 2(1) (2005) 109-114.
7. Li, W.B., Liu, C.N., Chen, Y.Y.: Combining multiple email filters of Naive Bayes based on GMM. *ACTA ELECTRONICA SINICA*. 34(2) (2006) 247-251.
8. Jos, M.G.H., Manuel, M.L., Enrique, P.S.: Combining text and heuristics for cost-sensitive spam filtering. In: Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning, Vol. 7 (2000) 99-102.
9. Segal, R., Crawford, J., Kephart, J. et al.: SpamGuru: An enterprise anti-spam filtering system. In: Proceedings of the First Conference on Email and Anti-Spam, (2004).
10. Hardle, W., Simar, L.: *Applied Multivariate Statistical Analysis*. 341-357 (2003).
11. Lewis, D.D.: Naive (Bayes) at forty: The independence assumption in information retrieval. In: Proceedings of the 10th European Conference on Machine Learning, LNCS 1398, Heidelberg. Springer (1998) 4-15.
12. McCallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: Proceedings of AAAI-98 Workshop on Learning for Text Categorization, (1998) 41-48.
13. PU1 and Ling-Spam dataset: <http://vit.demokritos.gr/skel/i-config/downloads/>.
14. Yang, Y., Pedersen, J.O.: 1997. A comparative study on feature selection in text categorization. In: Proceedings of the 14th International Conference on Machine Learning, (1997) 412-420.

Employee Profiling in the Total Reward Management

Silverio Petruzzellis¹, Oriana Licchelli², Ignazio Palmisano²,
Valeria Bavaro³, and Cosimo Palmisano³

¹ Cézanne Software S.p.A. - Via Amendola 172/C, I70126, Bari, Italy
silverio.petruzzellis@Cezannesw.com

² Dipartimento di Informatica - Università di Bari - Via E. Orabona 4, I70126, Bari, Italy
{licchelli, palmisano}@di.uniba.it

³ DIMeG, Politecnico di Bari - Viale Japigia 182, I70100 Bari, Italy
valeriabavaro@yahoo.it, c.palmisano@poliba.it

Abstract. The Human Resource departments are now facing a new challenge: how to contribute in the definition of incentive plans and professional development? The participation of the line managers in answering this question is fundamental, since they are those who best know the single individuals; but they do not have the necessary background. In this paper, we present the *Team Advisor* project, which goal is to enable the line managers to be in charge of their own development plans by providing them with a personalized and contextualized set of information about their teams. Several experiments are reported, together with a discussion of the results.

1 Introduction

Total reward management (TRM) is a holistic practice which offers a new action mode oriented towards pushing the use of IT in supporting the improvement of people performances, by understanding employee needs and by designing customized incentives and rewards. The Human Resource (HR) departments, that historically take care of the personnel development issues, are now becoming a more strategic component for the company, by contributing in the definition of incentive plans and professional development. The participation of the line managers in the process is fundamental, since they are those who best know the single individuals, working side by side and coordinating their activities. On the other hand they do not possess the right background to cope with internal conflicts, to make objective evaluations and to act as compensation experts towards their staff. They need the right support and a good mix of up-to-date information in order to apply their judgement in a timely and effective manner. The *Team Advisor* project addressed those issues by designing and implementing a platform for collaboration in the complex area of personnel development and reward planning. The application of profiling methodologies allowed to create personalized incentive plans that could reflect the history and the habits of the single individuals as well as balancing the fulfillment of her or his needs with the company's global perspective.

A recent report from OD&M Consulting [1] reveals that the actual trend is to include within Total Rewards System those benefits that satisfy individual specific requirements about wealth, welfare, security and mobility, whilst pay meets financial needs. To be effective, a personalized rewarding system must focus on the positive outcomes for

the organization, mostly depending on the approach chosen to manage this innovative change. Therefore we conducted a market analysis on the actual context of software applications¹ used for supporting HRM. The market segmentation was conducted using *variety of rewards offer* and *focus on profiling techniques* as segmentation variables. Which are essential requisites to create customized reward recommendations. None of the assessed information systems presented a good mix of both required features. Most of the applications commercially available have been developed essentially for supporting personnel administration, with no attention to more complex processes, manifesting an insufficient level of detail in the rewards range definition. The research we conducted has shown that there is a lack of applications strengthened by those decision support functionalities for simplifying the planning of reward interventions and for being receptive to employee needs. Therefore, the ideal placement of the envisioned software application would combine a high-level quality in profiling technologies and in reward library definition, together with a strong interconnection between the two aspects. More accurately, the purpose is to create a kind of *virtual manager* supporting enterprise compensation and reward policies.

2 Team Advisor Project

Within the Team Advisor project, our work was focused on the application of the User Profiling techniques in the TRM context, where a user profile contains information about needs and interests (part of the organization incentive programmes). In the Team Advisor project, the recommendations generated by the system are addressed to the line manager, suggesting the best reward policy for a specific staff member; the line manager is then in charge of finding the right balance between individual expectations and the company reward strategy.

The profile is built upon both demographic data and behavioral preferences collected through a commercial HRM portal, taking into account privacy issues: users wish to have their privacy guaranteed and they are also afraid of being controlled. One possibility is to handle personal data for the customer so that he can control/edit his own profile; in addition, the company must guarantee not to give personal data to a third party. Therefore, the core issues are trust and transparency, because the user wants the companies to respect their privacy by clearly declaring when they are collecting the personal data, offering opt-in to data collection programs rather than opt-out only choices (see EU Privacy Directive [2]).

2.1 Learning Employee Profiles Through the Personalization Engine

The Employee Profiling system is used to create profiles of the employees. It is based on a Personalization Engine (PE) [3,4] that uses information stored in a commercial

¹ Authoria Inc., www.authoria.com/AdvisorSeries.204.aspx, Nexcentec Software Inc., www.nexcentec.com, Axiom Software Ltd., www.axiomsoftware.com/discprofile.asp, Onesis S.p.A, www.onesis.it, ExecuTRACK Software AG, www.executrack.com/en/solutions.html, EBC Consulting Srl, www.ebcconsulting.com/cgi-bin/private/hrms.area.cgi, TPC&Join s.r.l., www.idipendenti.it/demo/demo.asp, Mizar Informatica s.r.l., www.mizar.it/prodotti/index.php#minosse

```

if Seniority <= 7.0 and Job Class = Employee-CONS then
  Class: yes
else if Seniority > 7.0 then
  Class: Estate
else if Job Class = White Collar-SALES then
  Class: no
Otherwise Class: Saloon

```

Fig. 1. An example of 4 classification rules for the reward category *Company Car*



The screenshot shows the 'Cognitive' application interface. At the top, there are navigation tabs: 'My Data', 'My Development', 'Team Management', and 'Team Development'. Below these, the current view is 'Team Management > Basic Data: William Lurs'. On the left, there is a sidebar with the 'Cognitive' logo and a list of employee details: Full Name (William Lurs), E-Mail (hlurs@acme.com), Company (HCP US), Location (Atlanta), Job (Credit Analyst), Organizational Unit (Atlanta Branch), and Main Position (Branch Manager Assistant). On the right, there is a table titled 'Recommended Rewards' with columns for Reward Name, Label, and Ratings. The table lists various rewards and their corresponding labels and ratings.

Reward Name	Label	Ratings
audio-visual media	no	1
holiday house	no	1
notebook	yes	0.75
free tickets for cultural events	no	1
company car	saloon	0.666
flexitime	no	0.9
handheld	no	1
agreement with transport enterprises	yes	0.823
reductions on product or company services	no	0.75
refund of fuel	yes	1

Fig. 2. Recommendations for an employee

Human Capital Management application. PE uses supervised learning techniques to automatically discover users' preferences. Through the WEKA library [5], it induces rule sets for classification. Input and output of PE are RDF models. These models contain: learning process input (positive and negative examples), definition of the learning problem (features and possible values), learning process output (classification rules) and classification phase input, in which the learned rules are used to categorize previously unseen instances; finally, they contain classification results.

In the context of TRM, the problem of learning employee's preferences can be cast to the problem of inducing concepts from examples labelled as members (or non-members) of the concepts. Given a finite set of categories of rewards $C = \{c_1, c_2, \dots, c_n\}$, the task consists in learning the target concept T_i employees interested in c_i . In the training phase, each individual set of features represents a positive example regarding the categories the employee is interested in and a negative example for other categories. The subset of the instances chosen to train the learning system has then to be labelled by a domain expert, that classifies each instance as member or non-member of each category. In this project, in order to facilitate the definition of the training sets, we introduced a pre-classification process working on population samples instead of single individuals. This approach allowed the generation of training sets without the direct knowledge of the candidates, thus taking advantage of a natural clustering habit of the expert user.

The training instances were processed by PE, which induced a classification rule set for each reward category (Figure 1) by working on the whole set of features. The

learning algorithm adopted in the profile generation process is based on PART [6], a rule-based learner that produces rules from pruned partial decision trees. The actual employee profile generation process is performed by applying this set of rules to the employee data. For each reward category, the system is able to predict whether the employee is interested in it or not, specifying the hypothetical satisfaction level for each. An employee profile, as intended within the Team Advisor project, is then composed by two main frames: the frame of employee data, and the frame of the rewards *learned* by the system.

3 Exploiting the Profiles to Support the Manager Advisor

The reward customization can be managed experimenting a partial involvement of the line managers. They know their own colleagues and also can see the team as a whole, ensuring impartial reward assignment; moreover, the line manager gets enough empowerment and responsibility to make the HR decentralization concrete, permitting a useful reduction of the time to decision, without great expenses in the implementation of the procedures [7]. However, they could feel uncomfortable in assigning rewards not being HR experts. The philosophy of the proposed IT solution fits the line devolution constraints: the use of a collaborative software platform to build a preliminary reward plan supports decision-making effectively. Line managers can then add their judgement coming from direct experience. To substantiate these reflections with a use case, we created a demo data set representing a US firm working in the banking/financial industry, and did a full run of the Team Advisor system within the company reward assignment process. An HR domain expert sets up the employee profile generation process in two phases. First, he selects a list of the most representative data about identity, professional path and socio-demographical background of the employees. Personal and socio-demographical data closely pertain to human dimension of the employee and are the same for every company. They can be collected and updated through periodical on line surveys. On the contrary, data about job conditions are context-sensitive, and may be more or less relevant according to the external environment and to the company policy. Second, he selects the rewards listed into the organizational incentive plan (such as: audio-visual media, flexitime, company car, reductions on product or company services, etc.) to build the Team Advisor Reward Library (created following a OD&M study [1]). For each item a domain is also specified (yes/no labels or multivalued labels, i.e. *saloon*, *estate*, *city car*, as possible values for the item *company car*). A user profile generation process is then started, and it produces the employee profiles as previously described. As an example, let us consider Fred Sidel from the Atlanta Branch, one of the managers of our sample organization, receiving recommendations about potential rewards to be assigned to his team, through a dedicated section of his digital dashboard. The system provides him with a detailed view of the recommendations for each of his thirteen coworkers, as shown in figure 2. For each recommendation one single label is specified, that is the domain value for which the recommender has computed the highest popularity rating; the goal of recommendations is to inform the line manager about the rewards and their adherence with the employee profile. It's up to the manager then to decide whether to use such information, including the reward recommendation into the

incentive plan for each employee, or ignoring it and adopting an autonomous evaluation method.

4 Experimental Session

During the testing phase of the Team Advisor software the aim was to measure the reliability of the recommendations generated by the model and, at the same time, to receive a feedback about their quality and usability by the line managers. The experimentation was structured as a role playing game, involving some of the managers of Cézanne Software. They claimed to be some of the managers populating the training set we previously described and they had to manage teams of 4-5 people also chosen within the available dataset. This condition helped us to ensure all the managers to approach the team in the same way, reducing the influence of familiarity on decision making. A short employee profile of all the employees involved was made available to each manager, together with the list of all the incentives of our Reward Library. In the first step the manager chose, among the reward library options, a maximum of 3 incentives for each employee. The choice could be both a *positive assignation*, if he thought that employee really liked to receive a certain reward, or a *negative assignation*, if he suggested avoiding the assignation of a specific disliked reward. Comparing manager selections with the reward recommendations generated by the system about each employee, we discovered that the number of assignations where managers and recommender completely disagree is quite high. This is probably due to the fact that involved people based their decisions only on the short data set provided, and this might compromise a correct process of reward selection. In a real context, in fact, the employee profile only represents an added element for the manager, whose decisions come from matching human and personal components and objective ones.

In the second phase of the experimentation managers filled a short user-satisfaction survey which would help us to formulate guidelines for a further model development. The table 1 synthesizes the survey results: for each considered dimension the table reports the distribution of user answers. Globally, the managers were quite satisfied of the Team Advisor recommending process and in particular they appreciated the quantity and variety of suggestions provided for each employee. On the contrary, they seemed skeptic about the usefulness of the recommendations (intended as the number of

Table 1. The survey results

	Unsatisfied	Barely Satisfied	Indifferent	Satisfied	Very Satisfied
Overall Satisfaction			1	4	
Item Personalization			2	3	1
Item Variety				2	3
Item Number			1		4
Item Reliability		1	1	3	
Item Usefulness			2	3	
Support Provided to the Line Manager in Rewarding the Team			1	1	
	Unimportant	Barely Important	Indifferent	Important	Very Important
Transparency of the Assignation Process		1		4	

suggestions the managers considered consistent but unexpected or unusual): all of them disagree from re-using such unusual items for rewarding their team. This is probably due to the lack of knowledge about the matching process between incentives and profile features. All the interviewed, in fact, were interested in receiving more details about the algorithm used for the assignation process: therefore, the transparency of the association paths might increase the user trustworthiness toward the Team Advisor Software and stimulate the manager to search for new levers motivating the workgroup. Finally, the managers considered that the system would streamline their engagement in managing rewarding process, confirming that such a semiautomatic tool might effectively impact on a HRM devolution.

5 Conclusions and Future Works

In this paper we presented the project Team Advisor, that uses the User Profiling techniques in the TRM context to provide line managers with recommendations which reveal unexpressed connections among the various elements that guide the reward assignation process. Starting from the results achieved in the prototypical phase, two experimental sessions were performed to evaluate the overall satisfaction as well as the acceptance level of the recommendations w.r.t. their usefulness, clarity and completeness.

Future software developments include live collection of user feedback on recommendations, and use of the collected feedback to refine the profile generation process. This will enable the system to be a dynamic engine for employee profiling, thus supporting the organizations in refining and sharing the knowledge about their human capital.

References

1. OD&M Consulting: Rapporto benefits 2005 (2005) in collaborazione con Lavoro & Carriere - ed. Il Sole 24 Ore.
2. European Parliament and Council of the European Union: Directive 95/46/ec of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. In: Official Journal of the European Communities. (1995) 31
3. Licchelli, O.: Personalization in Digital Libraries for Education. Computer Science in the Graduate Division of the University of Bari, Italy (2005)
4. Semeraro, G., Abbattista, F., Degemmis, M., Licchelli, O., Lops, P., Zambetta, F.: Agents, personalisation and intelligent applications. In Corchuelo, R., Cortés, A.R., Wrembel, R., eds.: Technologies Supporting Business Solutions, Part IV: Data Analysis and Knowledge Discovery, Chapter 7. Nova Sciences Publishers (2003) 141–160
5. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, San Francisco (2000)
6. Frank, E., Witten, I.: Generating accurate rule sets without global optimization. In: Proceedings of the 15th International Conference on Machine Learning, Morgan Kaufmann (1998) 144–151
7. Whittaker, S., Marchington, M.: Devolving HR responsibilities to the line. Threat, Opportunity or Partnership? *Employee Relations* **25**(3) (2003) 245–261

Mining Association Rules in Temporal Document Collections

Kjetil Nørnvåg*, Trond Øivind Eriksen, and Kjell-Inge Skogstad

Dept. of Computer and Information Science, NTNU
7491 Trondheim, Norway
Kjetil.Norvag@idi.ntnu.no

Abstract. In this paper we describe how to mine association rules in temporal document collections. We describe how to perform the various steps in the temporal text mining process, including data cleaning, text refinement, temporal association rule mining and rule post-processing. We also describe the Temporal Text Mining Testbench, which is a user-friendly and versatile tool for performing temporal text mining, and some results from using this tool.

1 Introduction

Temporal document databases¹ can be used for efficient storage and retrieval of temporal documents. They also provide efficient support for queries involving both document contents and time [9]. However, in addition to the explicit information and knowledge that can be retrieved using these techniques, the documents also contain implicit knowledge inside particular documents, as well as inter-document and inter-version knowledge. In order to discover this knowledge, data mining techniques have to be applied. In this paper we describe how to mine association rules in temporal document collections. An example of such a rule from a web newspaper is “if one version of the page contains the word *bomb*, a subsequent version will contain the word *Al Quaida*”.

Finding temporal association rules in text documents can be useful in a number of contexts. Examples are 1) analysis of health records to find relationships between medicine, symptoms, and diseases, 2) investigations, and in general understanding impact of events in the real world.

The main contributions of this paper are 1) introducing temporal association rule mining in document databases, 2) identifying appropriate association rule models and algorithms for performing the association rule mining, 3) developing techniques for pre- and post-processing that increases the proportion of interesting rules, and 4) presenting some preliminary results from mining a web newspaper using the Temporal Text Mining (TTM) Testbench. It should be emphasized that mining text is quite different from mining structured data or retail transaction databases. The most important difference is the large number of itemsets/words (both number of distinct words and number of words in each “transaction”).

* Corresponding author.

¹ A temporal document database stores both previous and deleted document versions in addition to current non-deleted versions, and to each version is also kept the associated timestamp.

The organization of the rest of this paper is as follows. In Section 2 we give an overview of related work. In Section 3 we perform a study of variants of temporal rule mining in order to find an appropriate starting point for mining association rules in temporal document collections. In Section 4 we describe in detail the process of mining temporal association rules in temporal document databases. In Section 5 we describe the TTM Testbench, a tool for performing the text mining process, and in Section 6 we present some results from applying the TTM Testbench on a temporal document collection. Finally, in Section 7, we conclude the paper and outline issues for further work.

2 Related Work

Introduction to *data mining in general* can be found in many good text books, for example [2]. The largest amount of work in *text mining* have been in the areas of categorization, classification and clustering of documents, we refer to [1] for an overview of the area. Algorithms for mining association rules between words in text databases (if particular terms occur in a document, there is a high probability that certain other terms will occur in the same document) was presented by Holt and Chung in [3]. In their work, each document is viewed like a transaction, and each word being an item in the transaction.

Much research has been performed on aspects related to temporal data mining, and a very good survey of temporal knowledge discovery paradigms and methods is given by Roddick and Spiliopoulou [10]. As will be described in more detail in the rest of the paper, of particular interest in the context of our work is research in intertransaction association rules. The first algorithms for finding intertransaction rules described in the literature, E-Apriori and EH-Apriori[7], are based on the Apriori algorithm. These are extensions of the Apriori algorithm, where EH-Apriori also includes hashing. A further development of intertransaction rules is the FITI algorithm [12], which is specifically designed for efficient mining intertransaction rules.

A general problem in mining association rules is the selection of interesting association rules within the overall, and possibly huge set of extracted rules. Some work in this area exist, either based on statistical methods [11] or by considering the selection of association rules as a classification task [4].

Related to our work is trend analysis in text databases, where the aim is to discovery increasing/decreasing popularity of a set of terms [6,8]. A variant of temporal association rule mining is taking into account the exhibition periods of items [5].

3 Strategy for Mining Association Rules in Temporal Document Databases

In temporal association rule mining one tries to adopt the traditional association rule mining to cover temporal relations. Different approaches have been conducted to try and utilize this extra temporal information. These approaches can be divided into five categories, *episode rules*, *trend dependencies*, *sequence rules*, *calendric rules* and *inter-transaction rules* [2]. We will in the rest of this section describe the various approaches and an analysis of their usefulness for mining rules in temporal document collections.

3.1 Strategies

Episode Rules. Episode rules are a generalization of association rules. Rules are here applied to sequences of events, each event occurring at a particular time. An episode is further a sequence of events in a partial order and an episode rule is defined as an implication on the form $\mathcal{A} \Rightarrow \mathcal{B}$, where $\mathcal{A}$ and $\mathcal{B}$ are episodes and $\mathcal{A}$ is a subepisode of $\mathcal{B}$. This technique can be used in for example predicting certain events by sequences of earlier events.

With regards to textual data this approach will produce rules where for example each word or concept is mapped to an event. A result of this can be the episode rule $\{Bush, Iraq\} \Rightarrow \{Bush, Iraq, UN\}$, predicting the word *UN* based on the words *Bush* and *Iraq*.

This technique requires that the data analyzed is represented as a sequence of events. This can be achieved with textual data, by mapping each event to a document. However, this only creates rules describing relations between whole documents. Finding rules describing relations between concepts is more difficult. If each document is represented by a single concept the task is trivial, however this severely limits the possible rules that can be found.

Sequence Rules. Sequence rules are much like episode rules and share many of their properties. Similar to episode rules, sequences are used to describe temporal relations, but they differ in the representation of the dataset. Instead of a single sequence, sequence rules use transactions like in the traditional association rules. In addition these transactions are grouped by a common concept so that rules can be found related to that concept.

An example of a sequence rule is $\{AB, A\} \Rightarrow \{C, A\}$, where $\mathcal{A}$ and $\mathcal{B}$ appears at the same time followed by $\mathcal{A}$, implies a sequence where $\mathcal{C}$ is followed by $\mathcal{A}$. The transactions in the example are grouped by a common concept. Using textual data, the same kind of rules can be found if one can map the documents to this representation. For example, can *Author : noervaag* be used as grouping concept and each transaction represent an article produced by this author. Rules may then be found describing for example the evolution of writing patterns. This technique is not applicable if no such groupable concept is available.

Calendric Rules. With calendric association rules a predefined time unit and a calendar, given by a set of time intervals, are needed. This technique identifies rules within the predefined time intervals, which allows for seasonal changes to be analyzed. This also allows for the user to specify which time intervals he or she is most interested in. For example the user may specify the time unit to be day and time intervals by day number, for example $\{(1, 31) (334, 365)\}$. Here, the user is interested in rules describing patterns within January and December. The rules found is in the form $\mathcal{A} \Rightarrow \mathcal{B}$, like in traditional association rule mining, but restricted to the chosen calendar.

This approach is similar to finding traditional association rules with a limited dataset used, containing only transactions from the specified time interval. The user has with this approach more freedom in specifying the time intervals he or she is interested in. The technique does however not produce rules describing relations between items with different timestamp. That is, traditional rules are found, only limited by the calendar.

For this reason, calendric rules do not cover the patterns this project aims to find, and will therefore not be studied further.

Trend Dependencies. Trend dependencies differ from the other approaches in that the attributes that are used have to be ordinal. That is, it has to be possible to create a totally ordered set of values of each attribute.

A trend dependency rule is a rule $\mathcal{A} \Rightarrow \mathcal{B}$, where $\mathcal{A}$ and $\mathcal{B}$ are patterns over a specific schema. These patterns are sets of items on the form (A, Θ) , where A is a reference to a specific attribute and Θ is an element in $\{<, =, >, \geq, \leq, \neq\}$. Trend dependencies can for example be used to discover patterns like: *an employee's salary always increases over time*.

With regards to textual data, the requirement of ordinal values makes this approach less interesting. It may be possible to order some of the words or concepts that are used, but most likely this will not be the case with all of them.

Intertransaction Rules. Much of the literature focus on intratransaction association rules, which deal with relations within transactions. Only a few algorithms exist to mine intertransaction association rules, where relations across transactions (each having a timestamp) are analyzed. These algorithms usually utilize a time window to minimize computation.

Using an appropriate algorithm for finding intertransaction association rules, we can find rules on the form “*car* at time 0 and *hotel* at time 1 implies *leasing* at time 4”. As can be seen, these algorithms produce rules with items from different transactions given by a timestamp. The benefits with this approach is that here you not only know that the itemset $\mathcal{B}$ follows the itemset $\mathcal{A}$, but you also get an indication on when this is supposed to happen. This can be viewed as a significant increase in potential for describing temporal patterns.

Conclusion. Based on the analysis of the different approaches, we consider the use of intertransaction rules as most appropriate for our task. An efficient algorithm for mining intertransaction association rules will be described in Section 4.3.

4 Text Mining Process

In this section we describe in more detail the process of mining temporal association rules in temporal document databases. The process can be divided into the following steps, each possibly consisting of a number of sub-steps as will be described shortly: 1) tokenization, 2) text filtering and refinement, 3) mining temporal association rules based on the terms resulting from the text-refinement step, and 4) extracting the *interesting* association rules.

4.1 Tokenization

In this step the terms are extracted from the text documents. Depending on the application domain, terms that are numeric values may or may not be kept for mining. This step is a mapping from documents to a list of list of terms.

4.2 Text Filtering and Refinement

Documents can be large and each contain a large number of distinct terms. In order to increase quality of rules as well as reduce the computational cost, in general only a subset of the terms in the documents are actually part of the rule mining process. In the text filtering and refinement step it is determined which of the terms that shall participate, and certain transformation may also be performed in order to increase the probability of useful rules. This step consists of a number of operations, each having as input a list of list of terms and producing a new list of list of terms. It should be noted that the operations can be executed in different order than how they are presented below, and that not all have to be used in one rule mining process.

Text Filtering. The goal of text filtering is to remove terms that can be assumed to not contribute to the generation of meaningful rules. This step is vocabulary based, and the operations can be classified into two categories:

- Stop-word removal: Removing terms known to not be interesting or too frequently occurring, i.e., traditional stop-word removal. These terms are in general kept in a stop word list which contains terms that are considered stop words, but it might be that domain-related stop words are added as well.
- Finding interesting terms: Keeping only terms that are known to be good candidates for interesting rules. These terms are found in a particular vocabulary, which can either be general or domain-specific. In addition to domain-specific vocabularies, it is also possible to use more general vocabularies for this purpose, for example based on *nouns* or *proper nouns* (i.e., names of unique entities, in many languages like for example English these nouns start with a capital letter).

Text Refinement. In order to reduce the number of terms as well as increasing quality of the contributing terms, various approaches to text refinement can be employed. We have studied the use of the following techniques:

- Stemming, which determines a stem form of a given inflected (or, sometimes, derived) word form generally a written word form. This means that a number of related words all will be transformed into a common term.
- Semantic analysis of the text can be used to combine expressions (two or more terms) into one. One simple example of such combination is *United* and *States* into *United States*.

Term Selection. The goal of text refinement is to find those terms that are assumed to contribute most to giving meaningful rules. In order to increase quality of rules as well as reduce the computational cost, only a subset of the k terms in each document are actually part of the rule mining process. In the term selection step, the terms are selected based on their expected importance. In our work we have studied the use of two term-selection techniques: a) using the k first terms in a document, and b) using the k highest ranked terms based on the TF-IDF (term-frequency/inverse document frequency) weight of each term. The first technique is in particular useful for document collections where each document contains an abstract of the contents in the rest of the

document. However, the latter is most appropriate for general document collections. It should be noted that there is a danger of filtering out terms that could contribute to interesting rules when only a subset of the terms are used, so the value of k will be a tradeoff between quality and processing speed.

4.3 Rule Mining Using the First Intra Then Inter (FITI) Algorithm

In order to find intertransaction association rules (see Section 3.1), we employ the FITI algorithm [12]. The FITI algorithm for mining intertransaction rules is based on the property that *a large intertransaction itemsets must be made up of large intratransaction itemsets*. That is, for an itemset to be large in intertransaction association rule mining it will also need to be large using traditional association rule techniques. This property can be utilized to reduce the complexity of creating intertransaction rules. If all large intratransaction itemsets are known in advance, the candidate generation task only needs to consider itemsets present in this set. The FITI algorithm is based on this way of thinking and involves three phases: 1) mining large intratransaction itemsets, 2) database transformation, and 3) mining large intertransaction itemsets. Due to space constraints we do not go into more details of the FITI algorithm here but refer the interested reader to [12].

4.4 Rule Post-processing

In text mining in general, and temporal text mining in particular, a very large number of association rules will be found. A very important and challenging problem is to find those that are *interesting*.

Similar to traditional intertransaction association rules, parameters like *item set size* and measures like *support* and *confidence* are also important when creating intertransaction item sets and selecting final rules. Unlike traditional rule mining where often as large as possible item sets are created, in mining rules in text usually the rules based on relatively small item sets are sufficient. It should also be mentioned that because of the number of distinct terms the typical minimum support for association rules in text databases can be relatively low.

One particular aspect of rule mining in text is that often a high support means the rule is too obvious and thus less interesting. These rules are often a result of frequently occurring terms and can partly be removed by specifying the appropriate stop words. However, many will remain, and these can to a certain extent be removed by specifying a maximum support on the rules, i.e., the only resulting rules are those above a certain minimum support and less than a certain maximum support. Another technique that can be used to remove unwanted rules is to specify *stop rules*, i.e., rules that are common and can be removed automatically.

5 The TTM Testbench

The Temporal Text Mining (TTM) Testbench is a user-friendly application developed in order to simplify experimenting with rule mining in temporal document collections. Text mining using the TTM Testbench is essentially a process consisting of three

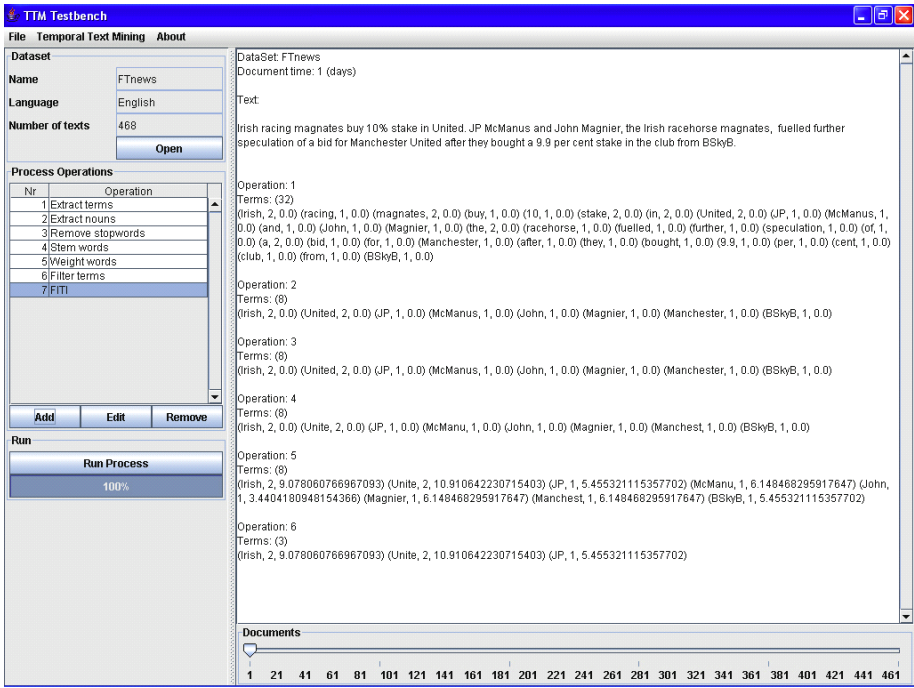


Fig. 1. TTM Testbench - After performing operations on the document collection

phases: 1) loading the document collection, 2) choosing which operations that should be performed, including pre- and post-processing as well as rule mining, and 3) let the system perform the selected operations and present the final result, i.e., the association rules generated based on the document collection, operations, and parameters.

Fig. 1 illustrates the TTM Testbench in use, after a document collection has been loaded. Studying the individual documents is also easy through the use of the scroll bar shown below the text window. To the left is the operation selection tool, where one or more operations are selected. It is also possible to view the result from the individual operations (for example extracted terms or TF-IDF values) for each document.

A number of operations are available in the TTM Testbench, each essentially working as part of a filtering/operator pipeline. The operators used in the experiments reported in this paper are:

Extract terms: Extract all terms in each document.

Extract nouns: Extract all proper nouns (i.e., nouns starting with a capital letter).

Remove stop words: Remove stop words. The stop words are stored in a stop word file where thee user himself add appropriate stop words.

Stem words: Perform stemming of the text. Both stemming of English and of Norwegian are supported.

Weight terms: Calculate a weight to each term, using TF-IDF.

Filter terms: Filter terms that should be included in the process when mining for temporal association rules. This operation has a number of options, the most important being the possibility of only keeping a certain number k of terms, which is either the k highest ranked based on TF-IDF value, or the k first words in the document.

FITI: The currently implemented algorithm for mining temporal association rules in the document collection that has been loaded into the system. For this operator a number of parameters are available, including minimum and maximum support, minimum and maximum confidence, the size of the time window, and maximum set size.

6 Experimental Evaluation

A number of experiments have been performed in order to validate the rule mining approach as well as studying the impact of various pre- and post-processing operations and rule mining parameters. We will in the following describe three of the experiments which have been performed using the TTM Testbench. The experiments are based on a relatively small collection, consisting of 38 days of the front page of the online version of Financial Times (468 news articles). The size of the dataset is relatively limited and it is therefore not expected that really interesting rules will be found using this dataset. However, it is believed that the dataset is large enough for identifying meaningful rules. Default parameters for the rule mining have been min/max support of 0.1/0.5, min/max confidence of 0.7/1.0, max time span 3 days, and max set size of 3.

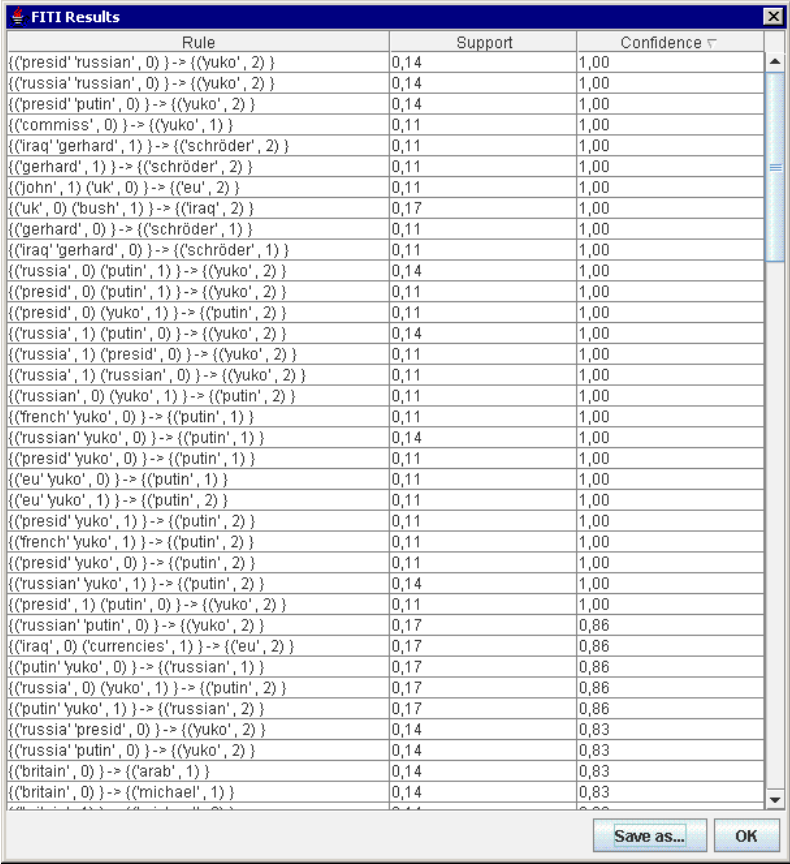
In order to determine the quality of the mined rules, evaluation criteria have to be defined. In these experiments focus will be on the quality of the rules found. To determine if a rule that has been mined is meaningful or not, is difficult (if not impossible) to determine by using automatic methods. For this project, a manual approach to determine rule quality is therefore used. Focus will be on finding rules that include terms with some semantic meaning, or that can be said to be self-explanatory, for example *Bush*, *Iraq* and *UN*. Automatic methods for determining if a rule is meaningful or not, are outside the scope of this paper. This should however be considered as a field for further research.

All terms. In the first experiment the only filtering performed is stop word removal and stemming. Stop words are removed in order to minimize the amount of rules that will not have any meaning, and stemming is included to group similar words and increasing the support of these. As can be expected, without limiting the number of terms there will be too many resulting rules. This result argues that some technique should be used to extract the semantics of the text and represent this by fewer terms, if rule mining is to be successful.

Filtering. Given the observation that using all terms is not feasible, a filtering operation can minimize the solution space. The most simple approach is to only use the first k terms of each text. However, the resulting rules do not carry much meaning. Examples of such rules include $(John, 0) \Rightarrow (hit, 1)$ or $(back, 0) \Rightarrow (Iraq, 1)$. This also points towards a more semantically-based approach.

An extension of using only the k first terms of each text is to use the k most important terms from each document. This is an attempt to reduce the number of features describing the text, without losing too much of the semantics. The TF-IDF-measure is used to determine the most important terms, and the k highest ranked terms are kept. Although the quality of the rules increased from the previous experiments, most of them were still not very meaningful.

Noun Extraction. Nouns have a high semantic meaning, and in order to see how this affects the rules we used proper nouns only as basis for the rule mining. Also in this experiment, filtering is performed. The reason for this is that there might be too many nouns present, making the approach unfeasible. In the result reported here the eight highest-weighted proper nouns from each document are used. Results of this experiment are shown in Fig. 2.



Rule	Support	Confidence τ
{{('presid' 'russian', 0)} -> {'yuko', 2}}	0,14	1,00
{{('russia' 'russian', 0)} -> {'yuko', 2}}	0,14	1,00
{{('presid' 'putin', 0)} -> {'yuko', 2}}	0,14	1,00
{{('commisss', 0)} -> {'yuko', 1}}	0,11	1,00
{{('iraq' 'gerhard', 1)} -> {'schroeder', 2}}	0,11	1,00
{{('gerhard', 1)} -> {'schroeder', 2}}	0,11	1,00
{{('john', 1)} -> {'eu', 2}}	0,11	1,00
{{('uk', 0)} -> {'iraq', 2}}	0,17	1,00
{{('gerhard', 0)} -> {'schroeder', 1}}	0,11	1,00
{{('iraq' 'gerhard', 0)} -> {'schroeder', 1}}	0,11	1,00
{{('russia', 0)} -> {'yuko', 2}}	0,14	1,00
{{('presid', 0)} -> {'yuko', 2}}	0,11	1,00
{{('presid', 0)} -> {'putin', 2}}	0,11	1,00
{{('russia', 1)} -> {'yuko', 2}}	0,14	1,00
{{('russia', 1)} -> {'presid', 0}}	0,11	1,00
{{('russia', 1)} -> {'yuko', 2}}	0,11	1,00
{{('russian', 0)} -> {'putin', 2}}	0,11	1,00
{{('french' 'yuko', 0)} -> {'putin', 1}}	0,11	1,00
{{('russian' 'yuko', 0)} -> {'putin', 1}}	0,14	1,00
{{('presid' 'yuko', 0)} -> {'putin', 1}}	0,11	1,00
{{('eu' 'yuko', 0)} -> {'putin', 1}}	0,11	1,00
{{('eu' 'yuko', 1)} -> {'putin', 2}}	0,11	1,00
{{('presid' 'yuko', 1)} -> {'putin', 2}}	0,11	1,00
{{('french' 'yuko', 1)} -> {'putin', 2}}	0,11	1,00
{{('presid' 'yuko', 0)} -> {'putin', 2}}	0,11	1,00
{{('russian' 'yuko', 1)} -> {'putin', 2}}	0,14	1,00
{{('presid', 1)} -> {'yuko', 2}}	0,11	1,00
{{('russian' 'putin', 0)} -> {'yuko', 2}}	0,17	0,86
{{('iraq', 0)} -> {'eu', 2}}	0,17	0,86
{{('putin' 'yuko', 0)} -> {'russian', 1}}	0,17	0,86
{{('russia', 0)} -> {'putin', 2}}	0,17	0,86
{{('putin' 'yuko', 1)} -> {'russian', 2}}	0,17	0,86
{{('russia' 'presid', 0)} -> {'yuko', 2}}	0,14	0,83
{{('russia' 'putin', 0)} -> {'yuko', 2}}	0,14	0,83
{{('britain', 0)} -> {'arab', 1}}	0,14	0,83
{{('britain', 0)} -> {'michael', 1}}	0,14	0,83

Fig. 2. Experimental results, nouns with stemming and the 8 first nouns kept

7 Conclusions and Further Work

In this paper we have described how to mine association rules in temporal document collections. We described how to perform the various steps in the temporal text mining process, including data cleaning, text refinement, temporal association rule mining and rule post-processing. We also described the TTM Testbench and some results from using this tool.

Future work includes the development of appropriate metrics for rule quality and develop new techniques for rule post-processing. Regarding the rule mining itself, it is obvious that it is a computationally very costly process, and more work should be performed on how to optimize this part of the process.

References

1. S. Chakrabarti. *Mining the Web - Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers, 2003.
2. M. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice Hall, 2003.
3. J. D. Holt and S. M. Chung. Efficient mining of association rules in text databases. In *Proceedings of CIKM'99*, 1999.
4. D. Janetzko, H. Cherfi, R. Kennke, A. Napoli, and Y. Toussaint. Knowledge-based selection of association rules for text mining. In *Proceedings of ECAI'2004*, 2004.
5. C.-H. Lee, C.-R. Lin, and M.-S. Chen. On mining general temporal association rules in a publication database. In *Proceedings of ICDM'2001*, 2001.
6. B. Lent, R. Agrawal, and R. Srikant. Discovering trends in text databases. In *Proceedings of KDD'1997*, 1997.
7. H. Lu, L. Feng, and J. Han. Beyond intratransaction association analysis: mining multidimensional intertransaction association rules. *ACM Trans. Inf. Syst.*, 18(4):423–454, 2000.
8. Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceedings of KDD'05*, 2005.
9. K. Nørvåg. Supporting temporal text-containment queries in temporal document databases. *Journal of Data & Knowledge Engineering*, 49(1):105–125, 2004.
10. J. F. Roddick and M. Spiliopoulou. Survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):750–767, 2002.
11. P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of KDD'2002*, 2002.
12. A. K. H. Tung, H. Lu, J. Han, and L. Feng. Efficient mining of intertransaction association rules. *IEEE Transactions on Knowledge and Data Engineering*, 15(1):43–56, 2003.

Self-supervised Relation Extraction from the Web

Ronen Feldman¹, Benjamin Rosenfeld¹, Stephen Soderland², and Oren Etzioni²

¹Computer Science Department, Bar-Ilan University
Ramat Gan, ISRAEL 52900

{feldman, brosenfeld}@cs.biu.ac.il

²Department of Computer Science, Washington University
Seattle, WA 98195-2350

{soderlan, etzioni}@cs.washington.edu

Abstract. Web extraction systems attempt to use the immense amount of unlabeled text in the Web in order to create large lists of entities and relations. Unlike traditional IE methods, the Web extraction systems do not label every mention of the target entity or relation, instead focusing on extracting as many different instances as possible while keeping the precision of the resulting list reasonably high. SRES is a self-supervised Web relation extraction system that learns powerful extraction patterns from unlabeled text, using short descriptions of the target relations and their attributes. SRES automatically generates the training data needed for its pattern-learning component. We also compare the performance of SRES to the performance of the state-of-the-art KnowItAll system, and to the performance of its pattern learning component, which uses a simpler and less powerful pattern language than SRES.

1 Introduction

Information Extraction [1-3] is the task of extracting factual assertions from text. Most IE systems rely on knowledge engineering or on machine learning to generate extraction patterns – the mechanism that extracts entities and relation instances from text. Both approaches require substantial human effort, particularly when applied to the broad range of documents, entities, and relations on the Web. In order to minimize the manual effort necessary to build Web IE systems, we have designed and implemented SRES (Self-Supervised Relation Extraction System). SRES takes as input the names of the target relations and the types of their arguments. It then uses a large set of unlabeled documents downloaded from the Web in order to learn the extraction patterns.

SRES is most closely related to the KnowItAll system developed at University of Washington by Oren Etzioni and colleagues [4], since both are self-supervised and both leverage relation-independent extraction patterns to automatically generate seeds, which are then fed into a pattern-learning component.

Our main contributions are as follows:

- We introduce a novel domain-independent system to extract relation instances from the Web with both high precision and relatively high recall.

- We show how we can integrate a simple NER component into the classification scheme of SRES in order to boost recall between 5-15% for similar precision levels.
- We provide an estimation of the true recall of SRES for unbounded relations such as merger and acquisition.
- We report on an experimental comparison between SRES, SRES-NER and the state-of-the-art KnowItAll system, and show that SRES can double or even triple the recall achieved by KnowItAll for relatively rare relation instances.

The rest of the paper is organized as follows: Section 2 describes previous work. Section 3 outlines the general design principles of SRES, its architecture, and then describes each SRES component in detail. Section 4 describes our extraction classification schema and Section 5 presents our experimental evaluation. Section 6 contains conclusions and directions for future work.

2 Related Work

The IE systems most similar to SRES are based on bootstrap learning: Mutual Bootstrapping [5], the DIPRE system [6], and the Snowball system [7]. Ravichandran and Hovy [8] also use bootstrapping, and learn simple surface patterns for extracting binary relations from the Web.

Unlike those unsupervised IE systems, SRES surface patterns allow gaps that can be matched by any sequences of tokens. This makes SRES patterns much more general, and allows recognizing instances in sentences inaccessible to the simple surface patterns of systems such as [5, 6, 8]. The greater power of SRES requires different and more complex methods for learning, scoring, and filtering of patterns. On the other hand, since our positive and negative examples are generated automatically we can not use more sophisticated patterns that include complex constraints utilized by fully supervised systems such as [9, 10].

Another direction for unsupervised relation learning was taken in [11]. This system uses a NER system to identify pairs of entities and then cluster them based on the types of the entities and the words appearing between the entities. Only pairs that appear at least 30 times were considered. The main benefit of this approach is that all relations between two entity types can be discovered simultaneously and there is no need for the user to supply the relations definitions. The precision reported by these systems (77% breakeven for the COM-COM domain) is inferior to that of SRES.

2.1 Brief Description of KnowItAll

KnowItAll uses a set of generic extraction patterns, and automatically instantiates rules by combining those patterns with user supplied relation labels. For example, KnowItAll has patterns for the generic “of” relation:

NP1 <relation> NP2

NP1 's <relation> , NP2

NP2 , <relation> of NP1

where NP1 and NP2 are simple noun phrases that extract values of attribute1 and attribute2 of a relation, and <relation> is a user-supplied string associated with the relation. Similar generic patterns are given for "direct-object" relations (such as acquisition) or symmetric relations (such as merger).

KnowItAll-PL. While those generic rules lead to high precision extraction, they tend to have low recall, due to the wide variety of contexts describing a relation. KnowItAll includes a simple pattern learning scheme (KnowItAll-PL) that builds on the generic extraction mechanism (KnowItAll-baseline). Like SRES, this is a self-supervised method that bootstraps from seeds that are automatically extracted by the baseline system.

3 Description of SRES

The goal of SRES is extracting instances of relations from the Web without human supervision. Accordingly, the input of the system is limited to (reasonably short) definition of the target relations (composed of the relation's schema and a few keywords that enable gathering relevant sentences). For example, this is the description of the acquisition relation:

Acquisition(ProperNP, ProperNP) ordered
keywords={"acquired" "acquisition"}

The word *ordered* indicates that *Acquisition* is not a symmetric relation and the order of its arguments matters. The *ProperNP* tokens indicate the types of the attributes. In the regular mode, there are only two possible attribute types – *ProperNP* and *CommonNP*, meaning proper and common noun phrases, respectively. When using the NER Filter component described in the section 4 we allow further subtypes of *ProperNP*, and the predicate definition becomes:

acquisition(Company, Company) ...

Additional keywords (such as “bought”, “purchased” etc), which can be used for gathering more sentences, are added automatically by using WordNet [12].

SRES consists of several largely independent components. The Sentence Gatherer downloads from the Web a large set of sentences that may contain target instances. The Seeds Generator uses a small set of generic patterns instantiated with the predicate keywords to extract a small set of high-confidence instances of the target relations. The Pattern Learner uses the seeds to learn likely patterns of relation occurrences. Then, the Instance Extractor uses the patterns to extract the instances from the sentences. Finally, the Classifier assigns the confidence score to each extraction.

3.1 Pattern Learning

The task of the *Pattern Learner* is to learn the patterns of occurrence of relation instances. This is an inherently supervised task, because at least some occurrences must be known in order to be able to find patterns among them. Consequently, the input to the Pattern Learner includes a small set (10 instances in our experiments) of known instances for each target relation. These seeds are generated automatically by the

generic patterns (described in Section 2.1) instantiated with the relation name and keywords. Those patterns have a relatively high precision (although low recall), and the top-scoring results, which are the ones extracted the highest number of times from different sentences, have close to 100% probability of being correct.

3.1.1 Preparing the Positive and Negative Sets

The positive set of a predicate (the terms *predicate* and *relation* are interchangeable in our work) consists of sentences that contain a known instance of the predicate, with the instance attributes changed to “<AttrN>”, where N is the attribute index. For example, assuming there is a seed instance *Acquisition(Oracle, PeopleSoft)*, the sentence “The Antitrust Division of the U.S. Department of Justice evaluated the likely competitive effects of Oracle's proposed acquisition of PeopleSoft” will be changed to “The Antitrust Division...of <Attr1>'s proposed acquisition of <Attr2>.”

The positive set of a predicate P is generated straightforwardly, using substring search. The negative set of a predicate consists of sentences with known false instances of the predicate similarly marked (with <AttrN> substituted for attributes). The negative set is used by the pattern learner during the scoring and filtering step, to filter out the patterns that are overly general. We generate the negative set from the sentences in the positive set by changing the assignment of one or both attributes to other suitable entities in the sentence. In the shallow parser based mode of operation, any suitable noun phrase can be assigned to an attribute. Continuing the example above, the following sentences will be included in the negative set:

<Attr1> of the <Attr2> evaluated the likely...
 <Attr2> of the U.S. acquisition of <Attr1>
 etc.

3.1.2 Generating the Patterns

The patterns for the predicate P are generalizations of pairs of sentences from the positive set of P . The function $Generalize(s_1, s_2)$ is applied to each pair of sentences s_1 and s_2 from the positive set of the predicate. The function generates a pattern that is the best (according to the objective function defined below) generalization of its two arguments.

The patterns are sequences of *tokens*, *skips* (denoted *), *limited skips* (denoted *?) and *slots*. The tokens can match only themselves, the skips match zero or more arbitrary tokens, and slots match instance attributes. The limited skips match zero or more arbitrary tokens, which must not have the types of any of the predicate attributes.

The $Generalize(s_1, s_2)$ function takes two sentences and generates the least (most specific) common generalization of both. The function does a dynamical programming search for the best match between the two patterns (Optimal String Alignment algorithm), with the cost of the match defined as the sum of costs of matches for all elements. The exact costs of matching elements are not important as long as their relative order is maintained. After the best match is found, it is converted into a pattern by copying matched identical elements and adding skips where non-identical elements are matched. For example, assume the sentences are

Toward this end, <Attr1> in July acquired <Attr2>
Earlier this year, <Attr1> acquired <Attr2> from X

Assuming that “X” belongs to the same type as at least one of the attributes while the other tokens are not entities, the best match will be converted to the pattern

*? *? *this* *? *? , <Attr1> *? *acquired* <Attr2> *? *

3.2 Post-processing, Filtering, and Scoring

The number of patterns generated at the previous step is very large. Post-processing and filtering tries to reduce this number, keeping the most useful patterns and removing the too specific, too general, and irrelevant ones.

First, we remove from patterns all “stop words” surrounded by skips from both sides, such as the word “*this*” in the last pattern in the previous subsection. Such words do not add to the discriminative power of patterns, and only needlessly reduce the pattern recall. The list of stop words includes all functional and very common English words, as well as punctuation marks. Note, that the stop words are removed only if they are surrounded by skips, because when they are adjacent to slots or non-stop words they often convey valuable information. After this step, the pattern above becomes

*? , <Attr1> *? *acquired* <Attr2> *

In the next step of filtering, we remove all patterns that do not contain *relevant* words. For each predicate, the list of relevant words is automatically generated from WordNet by following all links to depth at most 2 starting from the predicate keywords. For example, the pattern

<Attr1> * *by* <Attr2>

will be removed, while the pattern

<Attr1> * *purchased* <Attr2>

will be kept, because the word “*purchased*” can be reached from the keyword “*acquisition*” via synonym and derivation links.

The filtered patterns are then scored by their performance on the positive and negative sets. The scoring formula reflects the following heuristic: it needs to rise monotonically with the number of positive sentences it matches, but drops fast with the number of negative sentences it matches. Thus, the score of a pattern in our system is the number of positive sentences it matches, divided by the square of the number of negative sentences it matches plus 1:

$$\text{Score}(\text{Pattern}) = \frac{|\{S \in \text{PositiveSet} : \text{Pattern matches } S\}|}{(|\{S \in \text{NegativeSet} : \text{Pattern matches } S\}| + 1)^2}$$

4 Classifying the Extractions

The goal of the final classification stage is to filter the list of all extracted instances, keeping the correct extractions and removing mistakes that would always occur regardless of the quality of the patterns. It is of course impossible to know which extractions are correct, but there exist properties of patterns and pattern matches that increase or decrease the confidence in the extractions that they produce. Thus, instead

of a binary classifier, we seek a real-valued confidence function c , mapping the set of extracted instances into the $[0, 1]$ segment.

Since confidence value depends on the properties of particular sentences and patterns, it is more properly defined over the set of single pattern matches. Then, the overall confidence of an instance is the maximum of the confidence values of the matches that produce the instance.

Assume that an instance E was extracted from a match of a pattern P at a sentence S . The following binary features may influence the confidence $c(E, P, S)$:

- $f_1(E, P, S) = 1$, if the number of sentences producing E is greater than one.
- $f_2(E, P, S) = 1$, if the number of sentences producing E is greater than two.
- $f_3(E, P, S) = 1$, if at least one slot of the pattern P is adjacent to a non-stop-word token.
- $f_4(E, P, S) = 1$, if both slots of the pattern P are adjacent to non-stop-word tokens.
- $f_5(E, P, S) = 1$, if the number of nonstop words in P is 0 (f_5), 1 or greater (f_6), 2 or greater (f_7), 3 or greater (f_8), and 4 or greater (f_9).
- $f_{10} \dots f_{15}(E, P, S) = 1$, if the number of words between the slots of the match M that were matched to skips of the pattern P is 0 (f_{10}), 1 or less (f_{11}), 2 or less (f_{12}), 3 or less (f_{13}), 5 or less (f_{14}), and 10 or less (f_{15}).

As can be seen, the set of features above is small, and is not specific to any particular predicate. This allows us to train a model using a small amount of labeled data for one predicate, and then use the model for all other predicates. In cross-predicate tests we showed that classifier that performs well for all relations can be built using a small amount of labeled data for any particular relation.

Training: The patterns for a single model predicate are run over a relatively small set of sentences (3,000-10,000 sentences in our experiments), producing a set of extractions (between 150-300 extractions in our experiments).

The extractions are manually labeled according to whether they are correct or not.

For each pattern match $M_k = (E_k, P_k, S_k)$, the value of the feature vector $f_k = (f_1(M_k), \dots, f_{15}(M_k))$ is calculated, and the label $L_k = \pm 1$ is set according to whether the extraction E_k is correct or no.

A regression model estimating the function $L(f)$ is built from the training data $\{(f_k, L_k)\}$. For our classifier we used the BBR [13], but other models, such as SVM or NaiveBayes are of course also possible.

Confidence Estimation: For each pattern match M , its score $L(f(M))$ is calculated by the trained regression model. The final confidence estimates $c(E)$ for the extraction E is set to the maximum of $L(f(M))$ over all matches M that produced E .

NER Filter: In the SRES-NER version the entities of each candidate instance are passed through a simple rule-based NER filter, which attaches a score ("yes", "maybe", or "no") to the argument(s) and optionally fixes the arguments boundaries. The NER is capable of identifying entities of type PERSON and COMPANY (and can be extended to identify additional types).

The scores mean:

"yes" – the argument is of the correct entity type.

"no" – the argument is not of the right entity type, and hence the candidate instance should be removed.

"maybe" – the argument type is uncertain, can be either correct or not.

If “no” is returned for one of the arguments, the instance is removed. Otherwise, an additional binary feature is added to the instance's vector:

$f_{16} = 1$ iff the score for both arguments is “yes”.

For bound predicates, only the second argument is analyzed, naturally.

5 Experimental Evaluation

Our experiments aim to answer three questions:

1. What boost will we get by introducing a simple NER into the classification scheme of SRES?
2. How does SRES's performance compare with KnowItAll and KnowItAll-PL?
3. What is the true recall of SRES?

Our experiments utilized five binary relations: Acquisition, Merger, CEO_Of, MayorOf, InventorOf. The sentences for the experiments were collected by the KnowItAll crawler. The data for the Acquisition and Merger predicates consist of 900,000 sentences for each of the two predicates, where each sentence contains at least one predicate keyword. The data for the bounded predicates consist of sentences that contain a predicate keyword and one of a hundred values of the first (bound) attribute. Half of the hundred are frequent entities (>100,000 search engine hits), and another half are rare (<10,000 hits).

The pattern learning for each of the predicates was performed using the whole corpus of sentences for the predicate. For testing the precision of each of the predicates in each of the systems we manually evaluated sets of 200 instances that were randomly selected out of the full set of instances extracted from the whole corpus. In the first experiment we evaluate the full system over the whole dataset. In the second experiment we estimate the true recall of SRES.

5.1 Performance of the Whole System

In this experiment we compare the performance of SRES with classification to the performance of KnowItAll. To carry out the experiments, we used extraction data kindly provided by the KnowItAll group. They provided us with the extractions obtained by the KnowItAll system and by its pattern learning component (KnowItAll-PL). Both are sketched in Section 2.1 and are described in detail in [9]. The results are summarized in the four graphs in the Figure 1.

For three relations (Acquisition, Merger, and InventorOf (not shown due to lack of space)) SRES clearly outperforms KnowItAll. Yet for the other two (CEO_Of and MayorOf), the simpler method of KnowItAll-PL or even the KnowItAll-baseline do as well as SRES. Close inspection reveals that the key difference is the amount of redundancy of instances of those relations in the data. Instances of CEO_Of and MayorOf are mentioned frequently in a wide variety of sentences whereas instances of the other relations are relatively infrequent.

KnowItAll extraction works well when redundancy is high and most instances have a good chance of appearing in simple forms that KnowItAll is able to recognize. The additional machinery in SRES is necessary when redundancy is low. In the same graphs we can see that SRES-NER outperforms SRES by 5-15% in recall for similar precision levels.

5.2 True Recall Estimate

In this experiment we attempt to estimate the true recall of the system. It is impossible to manually annotate all of the relation instances because of the huge size of the input corpus. Thus, indirect methods must be used. We used a large list of known acquisition and merger instances (that occurred between 1/1/2004 and 31/12/2005) taken from the paid service subscription SBC Platinum. For each of the instances in this list we identified all of sentences in the input corpus that contained both instance attributes and assumed that all such sentences are true instances of the corresponding relation. This is of course an overestimate since in some cases the appearance of both attributes of a true relation instance is just a chance occurrence and does not constitute a true mention of the relation. Thus, our estimates of the true recall are pessimistic, and the actual recall is higher.

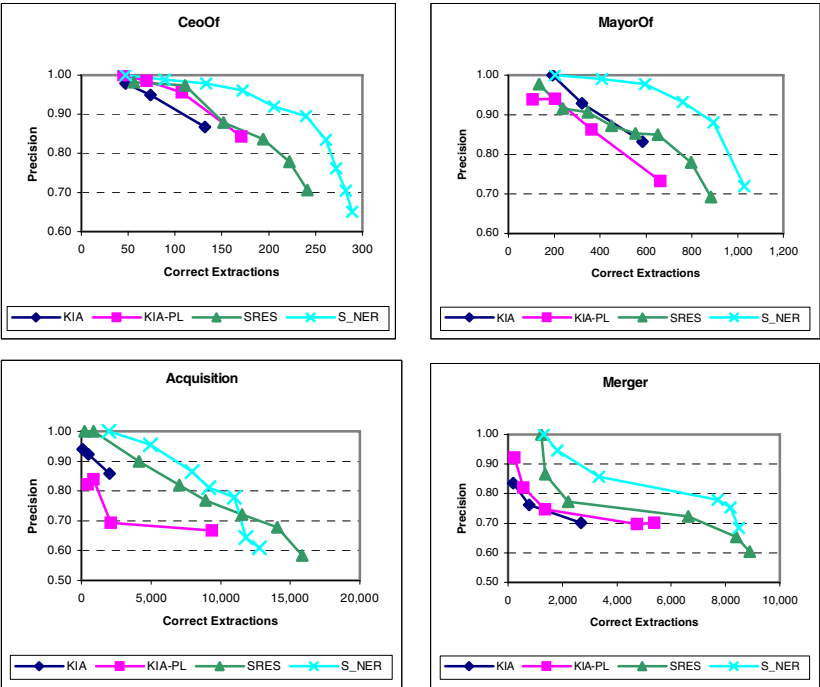


Fig. 1. Comparison between SRES, KnowItAll-baseline, and KnowItAll-PL

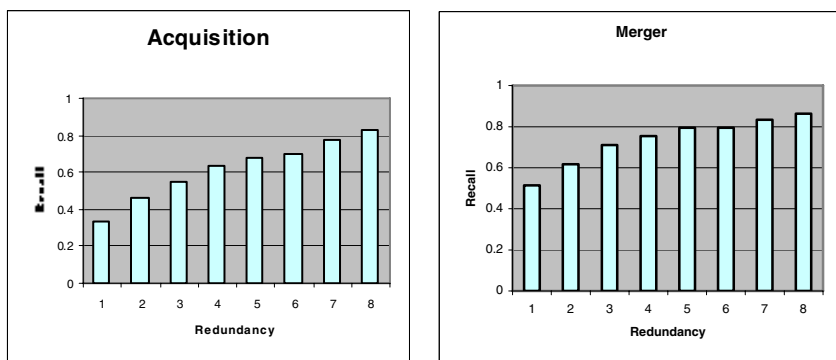


Fig. 2. True recall estimates

For all of the known true instances that appeared in the corpus we count the number of instances that were found by SRES. The fraction of found instances vs. all instances gives our recall estimate. In Figure 2 we present the results for the Acquisition and Merger relations. We show the recall values at several levels of redundancy. The redundancy of an instance in a corpus is the number of sentences in which both attributes of the instance appear. As can be seen from the graph, SRES identifies at least 40% of instances even at the lowest redundancy level.

6 Conclusions

We have presented the SRES system for autonomously extracting relations from the Web. We showed how to improve the precision of the system by classifying the extracted instances using the properties of the patterns and sentences that generated the instances and how to utilize a simple NER component. We performed an experimental comparison between SRES, SRES-NER and the state-of-the-art KnowItAll system, and showed that SRES can double or even triple the recall achieved by KnowItAll for relatively rare relation instances, and get an additional 5-15% boost in recall by utilizing a simple NER. In particular we have shown that SRES is more effective in identifying low-frequency instances, due to its more expressive rule representation, and its classifier (augmented by NER) that inhibits those rules from overgeneralizing.

References

1. Cardie, C., *Empirical Methods in Information Extraction*. AI Magazine, 1997. **18**(4): p. 65-80.
2. Cowie, J. and W. Lehnert, *Information Extraction*. Communications of the Association of Computing Machinery, 1996. **39**(1): p. 80-91.
3. Freitag, D. and A. McCallum, *Information Extraction with HMM Structures Learned by Stochastic Optimization*, in AAAI/IAAI. 2000. p. 584-589.
4. Etzioni, O., et al., *Unsupervised named-entity extraction from the Web: An experimental study*. Artificial Intelligence, 2005. **165**(1): p. 91-134.

5. Riloff, E. and R. Jones. *Learning Dictionaries for Information Extraction by Multi-level Bootstrapping*. in AAAI-99. 1999.
6. Brin, S. *Extracting Patterns and Relations from the World Wide Web*. in *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*. 1998. Valencia, Spain.
7. Agichtein, E. and L. Gravano. *Snowball: Extracting Relations from Large Plain-Text Collections*. in *Proceedings of the 5th ACM International Conference on Digital Libraries (DL)*. 2000
8. Ravichandran, D. and E. Hovy. *Learning Surface Text Patterns for a Question Answering System*. in *40th ACL Conference*. 2002.
9. Ciravegna, F. *Adaptive Information Extraction from Text by Rule Induction and Generalization*. in *In Proceedings of the 17th IJCAI 2001*. Seattle, WA.
10. Soderland, S., *Learning Information Extraction Rules for Semi-Structured and Free Text*. Machine Learning, 1999. **34**(1-3): p. 233-272.
11. Hasegawa, T., S. Sekine, and R. Grishman. *Discovering Relations among Named Entities from Large Corpora*. in *ACL 2004*. 2004.
12. Miller, G., *WordNet: An on-line lexical database*. International Journal of Lexicography, 1990. **3**(4): p. 235--312.
13. Genkin, A., D.D. Lewis, and D. Madigan, *Large-Scale Bayesian Logistic Regression for Text Categorization*. 2004, DIMACS: New Brunswick, NJ. p. 1-41.

Author Index

- Abe, Hidenao 379
Albayrak, Songül 29
Alberti, Marco 188
Alhajj, Reda 399
An, Aijun 493
Appice, Annalisa 369, 560
Ashish, Naveen 248
Avesani, Paolo 691
- Bantea, Ruxandra 290
Basile, Pierpaolo 707
Basile, Teresa Maria Altomare 258
Bavaro, Valeria 739
Bell, David 248
Berardi, Margherita 369
Berzal, Fernando 591
Bezuglov, Anton 642
Biba, Marenglen 258
Billot, Sylvie 697
Bosc, Patrick 284
Botta, Marco 409
Boulicaut, Jean-François 425
Bratko, Ivan 11
Buscicchio, Cosimo A. 38
- Cadoli, Marco 540
Calmet, Jacques 274
Caponetti, Laura 38
Carbonell, Jaime 167
Caroprese, Luciano 344
Castellano, Giovanna 208
Castiello, Ciro 208
Ceci, Michelangelo 560
Chesani, Federico 188, 338
Cho, Ming-Yuan 84, 580
Ciesielski, Krzysztof 713
Cleuziou, Guillaume 697
Cozzolongo, Giovanni 157
Crémilleux, Bruno 47
Cubero, Juan-Carlos 591
- d'Amato, Claudia 322, 570
Dardzińska, Agnieszka 445
Dau, Frithjof 332
de Bruin, Jeroen S. 419
- De Carolis, Berardina 157
De Matteis, Pietro 338
De Meo, Pasquale 147
Debenham, John 137
Degemmis, Marco 707
Dell'Acqua, Pierangelo 534
Demolombe, Robert 504, 514
Deslandres, Véronique 473
Di Mauro, Nicola 258
Doloc-Mihu, Anca 238
Durand, Nicolas 47
Dussauchoy, Alain 473
- Eichelberger, Christopher N. 671
Eklund, Peter 218
Elghazel, Haytham 473
Elomaa, Tapio 612
Eriksen, Trond Øivind 745
Esposito, Roberto 652
Etzioni, Oren 755
- Fanelli, Anna Maria 208
Fang, Fu-Min 84, 580
Fanizzi, Nicola 322, 570
Feldman, Ronen 755
Ferilli, Stefano 258
Franz, Thomas 1
Freire de Sousa, Jorge 632
- Garad, Akram Abu 296
Gartner, Alexandru 290
Gavanelli, Marco 188
Gawdiak, Yuri 248
Górecki, Przemysław 38
Görlitz, Olaf 1
Greco, Sergio 344
Grześ, Marek 121
Gu, Ja-Chul 435
- Haberdar, Hakan 29
Hacid, Mohand-Said 473
Hadžikadić, Mirsad 671
Han, Dongfeng 350
Hirano, Shoji 454
Hou, Young-Chang 58

- Hsieh, Chwen-Dar 681
 Hsu, Ching-Sheng 58
 Huang, Xiangji 493
 Hwang, Zion 178

 Immaneni, Trivikram 198

 Jin, Chun 167
 Jorge, Alípio M. 632
 Jouve, Pierre-Emmanuel 622
 Jung, Kyung-Yong 312

 Kaya, Mehmet 399
 Khashman, Adnan 296
 Kheddouci, Hamamache 473
 Kim, Gwanyeon 178
 Kim, June 19
 Kim, JungHo 19
 Kim, Yong 178
 Kłopotek, Mieczysław A. 713
 Krętowski, Marek 121
 Kujala, Jussi 612

 Lamma, Evelina 188
 Lawrence, Elaine 137
 Lee, Heungkyu 19
 Lee, Hong-Jen 84, 580
 Lee, Minsoo 178
 Lee, Sang-Chul 435
 Lee, Tsair-Fwu 84, 580
 Leo, Pietro 369
 Leone, Nicola 524
 Lesmo, Leonardo 550
 Lew, Stanislas 697
 Lewis, Rory A. 228
 Li, Ming 350
 Li, Wenbin 729
 Li, Wen-hui 68, 350
 Licchelli, Oriana 739
 Lietard, Nadia 284
 Lio, Vincenzino 524
 Lisi, Francesca A. 306
 Liu, Chao-Lin 601, 681
 Liu, Chunnian 729
 Liu, Dong-fei 68
 Liu, Jing 77
 Loglisci, Corrado 369
 Lombardi, Anna 534
 López de Mántaras, Ramon 18
 Lops, Pasquale 707

 Louis, Vincent 514
 Lu, Xiaosuo 350
 Lucas, Caro 91

 Maluf, David 248
 Mancini, Toni 540
 Marín, Nicolás 591
 Martin, Ben 218
 Martin, Lionel 697
 Mello, Paola 188, 338
 Memariani, Azizollah 91
 Meng, Yu 68
 Moniz Pereira, Luís 534
 Montali, Marco 338
 Moreira, João M. 632
 Možina, Martin 11

 Nicoloyannis, Nicolas 622
 No, Jaechun 268
 Nørvåg, Kjetil 745

 Ohsaki, Miho 379

 Palmisano, Cosimo 739
 Palmisano, Ignazio 739
 Panetta, Claudio 524
 Park, Chang Won 268
 Park, Sehyun 178
 Park, Sung Soon 268
 Patrizi, Fabio 540
 Pensa, Ruggero G. 425
 Petrescu, Diana 290
 Petruzzellis, Silverio 739
 Pivert, Olivier 284
 Pizzuttilo, Sebastiano 157
 Poletti, Nicola 691
 Polo, José-Luis 591
 Posea, Vlad 290
 Pozos Parra, Pilar 504
 Putz, Peter 248

 Quattrone, Giovanni 147

 Radicioni, Daniele P. 409, 652
 Raghavan, Vijay V. 238
 Rahgozar, Masoud 91
 Raś, Zbigniew W. 228, 445, 483
 Ritschard, Gilbert 463
 Robaldo, Livio 550
 Robardet, Céline 425

- Rosenfled, Benjamin 755
 Rousu, Juho 612
 Ruan, Chun 359
- Saathoff, Carsten 1
 Saitta, Lorenza 662
 Schenk, Simon 1
 Shen, Lincheng 101
 Shieh, Chin-Shiuh 84, 580
 Sizov, Sergej 1
 Skogstad, Kjell-Inge 745
 Soares, Carlos 632
 Soderland, Stephen 755
 Sona, Diego 691
 Song, Ohyoung 178
 Staab, Steffen 1
 Stefanowski, Jerzy 723
 Storari, Sergio 338
 Suh, Yung-Ho 435
 Sun, Jun 77
 Suzuki, Einoshin 47
 Synak, Piotr 389
- Terracina, Giorgio 524
 Thirunarayan, Krishnaprasad 198
 Thomas, Julien 622
 Tian, Jing 101
 Trausan-Matu, Stefan 290
 Trubitsyna, Irina 344
 Tsay, Li-Shiang 483
 Tsumoto, Shusaku 379, 454
 Tu, Shu-Fen 58
- Uhm Yoonsik 178
 Ursino, Domenico 147
- Varadharajan, Vijay 359
 Vargas, Juan E. 642
 Veeramachaneni, Sriharsha 691
 Vivaracho, Carlos E. 111
 Vrain, Christel 697
- Wang, Hao 127
 Wang, Jian-yuan 68
 Wang, Yi 350
 Wieczorkowska, Alicja 389
 Wu, Xindong 127
- Xu, Wenbo 77
- Yamaguchi, Takahira 379
 Yang, Yi 274
 Yao, Hongliang 127
 Yao, Qingsong 493
 Yu, Kui 127
- Žabkar, Jure 11
 Zarnani, Ashkan 91
 Zhang, Xin 228
 Zhang, Zhen-hua 68
 Zheng, Yanxing 101
 Zhong, Ning 729
 Zienkiewicz, Marcin 723
 Zighed, Djamel A. 463
 Zumpano, Ester 344