Lecture Notes in Artificial Intelligence 3735

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Achim Hoffmann Hiroshi Motoda
Tobias Scheffer (Eds.)

Discovery Science

8th International Conference, DS 2005
Singapore, October 8 – 11, 2005
Proceedings

Springer

Series Editors

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

Volume Editors

Achim Hoffmann
University of New South Wales
School of Computer Science and Engineering
Sydney 2052 NSW, Australia
E-mail: achim@cse.unsw.edu.au

Hiroshi Motoda
Osaka University
Department of Advance Reasoning
Division of Intelligent System Science
Institute of Scientific and Industrial Research
8-1 Mihogaoka, Ibaraki, Osaka, 567 Japan
E-mail: motoda@sanken.osaka-u.ac.jp

Tobias Scheffer
Humboldt University Berlin
Department of Computer Science
Unter den Linden 6, 10099 Berlin, Germany
E-mail: scheffer@informatik.hu-berlin.de

Library of Congress Control Number: 2005933095

CR Subject Classification (1998): I.2, H.2.8, H.3, J.1, J.2

ISSN 0302-9743
ISBN-10 3-540-29230-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-29230-2 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11563983 06/3142 5 4 3 2 1 0

Preface

This volume contains the papers presented at the 8th International Conference on Discovery Science (DS 2005) held in Singapore, Republic of Singapore, during the days from 8–11 of October 2005.

The main objective of the Discovery Science (DS) conference series is to provide an open forum for intensive discussions and the exchange of new ideas and information among researchers working in the area of automating scientific discovery or working on tools for supporting the human process of discovery in science. It has been a successful arrangement in the past to co-locate the DS conference with the International Conference on Algorithmic Learning Theory (ALT). This combination of ALT and DS allows for a comprehensive treatment of the whole range, from theoretical investigations to practical applications. Continuing in this tradition, DS 2005 was co-located with the 16th ALT conference (ALT 2005). The proceedings of ALT 2005 were published as a twin volume 3734 of the LNCS series.

The International Steering Committee of the Discovery Science conference series provided important advice on a number of issues during the planning of Discovery Science 2005. The members of the Steering Commitee are Hiroshi Motoda, (Osaka University), Alberto Apostolico (Purdue University), Setsuo Arikawa (Kyushu University), Achim Hoffmann (University of New South Wales), Klaus P. Jantke (DFKI and FIT Leipzig, Germany), Massimo Melucci (University of Padua), Masahiko Sato (Kyoto University), Ayumi Shinohara (Tohoku University), Einoshin Suzuki (Yokohama National University), and Thomas Zeugmann (Hokkaido University).

We received 112 full paper submissions out of which 21 long papers (up to 15 pages), 7 regular papers (up to 9 pages), and 9 project reports (3 pages) were accepted for presentation and are published in this volume. Each submission was reviewed by at least two members of the Program Committee of international experts in the field. The selection was made after careful evaluation of each paper based on originality, technical quality, relevance to the field of discovery science, and clarity.

The Discovery Science 2005 conference had three types of presentations: long papers were presented in a plenary session; regular papers were presented in a short spotlight presentation to generate interest and a presentation during a poster session for intensive discussions and presentation of details; project reports were presented in a poster session to allow intensive discussion on ongoing work and interesting ideas that had not been developed to the same degree of maturity as long and regular papers.

The Carl Smith Award was presented this year for the first time in honor of Professor Carl Smith to the student author of the best paper in the Discovery Science conference authored or co-authored by a student. The prize of 555 Euro was awarded to Qianjun Xu for the paper entitled *Active Constrained Clustering by Examining Spectral Eigenvectors*.

This volume consists of four parts. The first part contains invited talks of ALT 2005 and DS 2005. Since the talks were shared between the two conferences, for the speakers invited specifically for ALT 2005 only abstracts are contained in this volume, while the full paper is found in the twin volume LNCS 3734 (the proceedings of ALT 2005). We were delighted that Gary Bradshaw (Invention and Artificial Intelligence), Vasant Honovar (Algorithms and Software for Collaborative Discovery from Autonomous, Semantically Heterogeneous, Distributed, Information Sources), Chih-Jen Lin (Optimization Issues in Training Support Vector Machines), Ross D. King (The Robot Scientist Project), and Neil Smalheiser (The Arrowsmith Project: 2005 Status Report) followed our invitation to present their work.

The second part of this volume contains the papers accepted as long papers (acceptance rate of less than 21%). The third part of this volume contains the regular papers, which were found to belong to the best 27% of all submissions. Finally, the fourth part of this volume contains the project reports; the total acceptance rate for all three paper categories sums to 37% of all submissions.

We are deeply indebted to the Program Committee members as well as their subreferees who had the critically important role of reviewing the submitted papers and contributing to the intense discussions which resulted in the selection of the papers published in this volume. Without this enormous effort, ensuring the high quality of the work presented at Discovery Science 2005 would not have been possible.

We also thank all the authors who submitted their work to Discovery Science 2005 for their efforts.

We wish to express our gratitude to the invited speakers for their acceptance of the invitation and their stimulating contributions to the conference.

Finally, we wish to thank everyone who contributed to make Discovery Science 2005 a success: the DS Steering committee, the ALT conference chairs, invited speakers, and last but not least Lee Wee Sun, the Local Arrangements Chair and his team of supporters.

October 2005

Achim Hoffmann
Hiroshi Motoda
Tobias Scheffer

Organization

Conference Chair

Hiroshi Motoda	Osaka University, Japan

Program Committee

Achim Hoffmann	University of New South Wales, Sydney, Australia (Chair)
Tobias Scheffer	Humboldt-Universität zu Berlin, Germany (Chair)
Jose Luis Balcázar	University of Catalunya, Spain
Elisa Bertino	Purdue University, USA
Wray Buntine	Helsinki Institute of Information Technology, Finland
Vincent Corruble	University of Pierre et Marie Curie, France
Manoranjan Dash	Nanyang Technological University, Singapore
Andreas Dress	Max Planck Institute for Mathematics in the Sciences, Germany
Sašo Džeroski	Jozef Stefan Institute, Slovenia
Tapio Elomaa	Tampere University of Technology, Finland
Eibe Frank	University of Waikato, New Zealand
Johannes Fürnkranz	Technical University of Darmstadt, Germany
João Gama	University of Porto, Portugal
Gunter Grieser	Technical University of Darmstadt, Germany
Fabrice Guillet	Ecole Polytechnique of the University of Nantes, France
Mohand-Said Hacid	University of Claude Bernard, Lyon, France
Udo Hahn	Jena University, Germany
Tu Bao Ho,	JAIST, Japan
Klaus P. Jantke	FIT Leipzig, Germany
Szymon Jaroszewicz	Technical University of Szczecin, Poland
Kristian Kersting	Universität Freiburg, Germany
Ross King	University of Wales, UK
Kevin Korb,	Monash University, Melbourne, Australia
Ramamohanarao Kotagiri	University of Melbourne, Australia
Stefan Kramer	TU München, Germany
Nicolas Lachiche	Univ. Strasbourg, France
Nada Lavrač	Jozef-Stefan Institute, Ljubljana, Slovenia
Aleksandar Lazarević	United Technologies Research Center, CT, USA
Jinyan Li	Institute for Infocomm Research, Singapore

Ashesh Mahidadia	University of New South Wales, Sydney, Australia
Michael May	Fraunhofer Institute for Autonomous Intelligent Systems, Germany
Katharina Morik	University of Dortmund, Germany
Ion Muslea	Language Weaver, USA
Lourdes Peña	Center for Intelligent Systems at the ITESM, Mexico
Bernhard Pfahringer	University of Waikato, New Zealand
Jan Rauch	University of Economics, Czech Republic
Domenico Saccà	University of Calabria and ICAR-CNR, Italy
Rudy Setiono	National University of Singapore, Singapore
Myra Spiliopoulou	Otto-von-Guericke University, Germany
Ashwin Srinivasan	IBM India, India
Einoshin Suzuki	Yokohama National University, Japan
Masayuki Takeda	Kyushu University, Japan
Kai Ming Ting	Monash University, Australia
Ljupčo Todorovski	Jozef Stefan Institute, Slovenia
Volker Tresp	Siemens AG, München, Germany
Alfonso Valencia	National Centre for Biotechnology, Spain
David Vogel	AI Insight, USA
Gerhard Widmer	Johannes-Kepler-Universität, Austria
Akihiro Yamamoto	Kyoto University, Japan
Mohammed Zaki	Rensselaer Polytechnic Institute, USA
Chengqi Zhang	University of Technology Sydney, Australia
Djamel A. Zighed	University of Lumiere, France

Local Arrangements

Lee Wee Sun	National University of Singapore, Republic of Singapore

External Reviewers

Mohammed Al Hasan
Alexandre Aussem
Hideo Bannai
Maurice Bernadet
Steffen Bickel
Remco Bouckaert
Agnès Braud
Ulf Brefeld
Michael Brückner
Robert D. Burbidge
Mario Cannataro
Narendra S. Chaudhari
Vineet Chaoji
Maria Luisa Damiani
Marko Debeljak
Damjan Demšar
Isabel Drost
Timm Euler
Tanja Falkowski
Feng Gao
Gemma C. Garriga
Rémy Gaudin
Vivekanand Gopalkrishnan
Andrea Gualtieri
Pietro H. Guzzo
Mounira Harzallah
Kohei Hatano
Phan Xuan Hieu
Lucas Hope
Daisuke Ikeda
Branko Kavsek
Gaelle Legrand
Lee Wee Sun
Remi Lehn
Peter Ljubic
Chuan Lu
Carlo Mastroianni
Igor Nai-Fovino
Luigi Palopoli
Esa Pitkänen
Dragoljub Pokrajac
Lothar Richter
Ulrich Rückert
Martin Scholz
Alexander K. Seewald
Zujun Shentu
Giandomenico Spezzano
Shyh Wei Teng
Evimaria Terzi
Nguyen Truong Thang
Julien Thomas
Salvatore Vitabile
Michael Wurst
Shipeng Yu
Bernard Ženko

Sponsoring Institutions

We wish to thank the Air Force Office of Scientific Research and the Asian Office of Aerospace Research and Development for their contribution to the success of this conference.

AFOSR/AOARD support is not intended to express or imply endorsement by the U.S. Federal Government.

The Carl Smith Award

Starting with this year, the “Carl Smith Award” is presented to the most outstanding paper written or co-authored by a student. The selection is made by the actual program committee of the Discovery Science conference. The award carries a scholarship prize of 555 Euro.

The decison to introduce this award has been proposed at the ALT/DS-business meeting of last year’s conference in Padua after remembering Carl Smith, who passed away on July 21, 2004 after a long and valiant battle with cancer, with a minute of silence. Subsequently, this decison has been happily approved by Patricia Smith.

Carl performed his undergraduated studies in Vermont and received his Bachelor of Science Degree from the University of Vermont in 1972. Then, he moved to State University of New York at Buffulo where he received his Ph.D. Subsequently, he was Assistant Professor of Computer Science at Purdue University. Then he was at the University of Maryland at College Park, where he got promoted to Associate and Full Professor. In 1993, Carl received the Habilitation degree from the University of Latvia in Riga. He is also one of the very few non-Latvian scientists who got elected to the Latvian Academy of Science.

Additionally, Carl spent several years as program manager at the National Science Foundation’s theoretical computer science program and continued to work for the National Science Foundation by working on programs and panel reviews for many years.

Carl also contributed to the computer science community as an editor of the International Journal of the Foundations of Computer Science, Theoretical Computer Science, and Fundamenta Informaticae.

The Discovery Science conference series is still a young one but many researchers remember Carl for a much longer time, because of his very active role in the algorithmic or computational learning communities.

Let us look back to 1986 when the 1st International Workshop on Analogical and Inductive Inference was held in Wendisch-Rietz near Berlin. This was the starting point of the first international conference series on learning theory which merged in 1994 with the Algorithmic Learning Theory series established in 1990. At this workshop Carl gave a talk “On the Inference of Sequence of Functions” (co-authored with Bill Gasarch) in which he developed a model of “learning how to learn.” Of course, by this time Carl was already well known through his work on comparison of identification criteria for machine inductive inference, his work on team learning, and the beautiful survey paper “Inductive Inference: Theory and Methods” (co-authored with Dana Angluin).

Besides the very fruitful scientific discussions we all enjoyed at this workshop, it was also the beginning or continuation of a lasting friendship many of us had with Carl which in turn led to many teams including Carl all over the world. These long and fruitful collaborations included leading groups from Japan, Latvia, Germany, USA, Australia, and Singapore among many countries. As a result, papers on query learning, on memory limitation, on learning with anomalies, on the complexity of inductive inference, on Barzdinš's conjecture, on procrastination, on mind change complexity as well as on a logic of discovery emerged.

Besides his regular papers, Carl contributed in many ways to the ALT and DS conference series by serving for their Program Committees and the DS Steering Committee, and by serving as local chair, as conference chair and arrangements as invited speaker.

He also chaired IFIP WG 1.4 on Computational Learning Theory and organized many funding to support, in particular, young scientists.

Since Carl Smith did so much for the ALT and DS conferences, his spirit, his contributions, his passion, and his ideas will be remembered and passed to the young generations by the "Carl Smith Award."

August 2005 Thomas Zeugmann

Table of Contents

Invited Papers

Regular Contributions - Long Papers

Regular Contributions - Regular Papers

Project Reports

Invention and Artificial Intelligence

Gary Bradshaw

Psychology Department, P.O. Box 6161,
Mississippi State University, MS 39762, USA
glb2@ra.msstate.edu

Abstract. Invention, like scientific discovery, sometimes occurs through a heuristic search process where an inventor seeks a successful invention by searching through a space of inventions. For complex inventions, such as the airplane or model rockets, the process of invention can be expedited by an appropriate strategy of invention. Two case studies will be used to illustrate these general principles: the invention of the airplane (1799-1909) and the invention of a model rocket by a group of high school students in rural West Virginia in the late 1950's. Especially during the invention of the airplane, inventors were forced to make scientific discoveries to complete the invention. Then we consider the enterprise of artificial intelligence and argue that general principles of invention may be applied to expedite the development of AI systems.

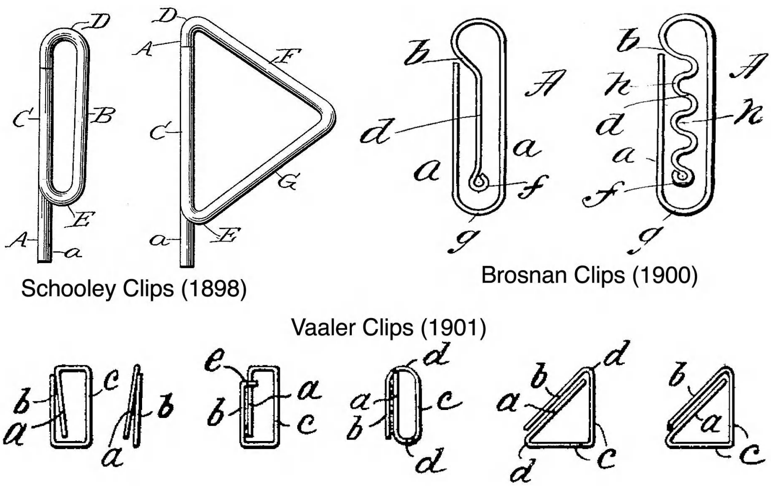
1 Heuristic Search and Invention

Humans live in a world that has been shaped by invention: the clothing we wear, the food we eat, our houses, our transportation, our entertainment – all depend on a vast aggregation of technology that has been developed over the millennia. Some invented artifacts, including stone knives and hammers, even predate homo sapiens. Given the importance of invention in the contemporary world, it is worth some effort to understand how new inventions are developed.

Even a superficial review of the history of technology and invention shows that many different paths can lead to an invention. Basalla [1] and Petroski [2] have discussed the similarities between biological evolution and technological invention. In reviewing several case studies, Basalla provides strong evidence that some inventions arise when inventors produce random mutations of existing inventions, and society determines which inventions are "fit" for reproduction. One example is the paper clip, which appeared about the same time as steel wire and wire-bending jigs became available. Figure 1 illustrates different clip shapes from three different American patents. The familiar double-oval Gem clip was never patented in the U.S., but other patents were filed that described wire loops of various shapes in an effort to create a clip that was easy to slip over a set of papers, did not tear the papers as it was used, and held the collection tightly.

A system of invention based on random mutation (by inventors) and natural selection (by society) does not appear to require any intelligent activity on the part of inventors themselves: how difficult can it be to bend wires, after all?

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 1–13, 2005.


Schooley Clips (1898)

Brosnan Clips (1900)

Vaaler Clips (1901)

Fig. 1. Various forms of paper clips patented in the U.S. around the turn of the century

But in spite of the variability of all of the patented paper clips, they all shared some important features in common. First, all clips are relatively flat, having a structure that is mostly two-dimensional rather than 3-dimensional. Next, clips tend to loop over themselves in a way that allows them to pinch together a stack of papers. If inventors were simply producing random bends in wire, they would most commonly produce non-planar bends that created a 3-dimensional structure and would most commonly produce forms that did not have the necessary loops to hold together a paper stack. In spite of the diversity of paper clip forms, it does seem clear that the various alternatives were not produced by blind or random mutation, but rather by strategic alteration, perhaps akin to genetic algorithms in use today.

It should also be evident that the "inventive play" described by Basalla, where inventors produce different forms through manipulation of an existing artifact, is not sufficient to account for inventions where a number of parts are necessary to the performance of the whole. Consider, for example, the television. Although it might be possible for an infinite monkey team to wire together tubes, resistors, capacitors, and transformers to produce a television set, it seems unlikely this would have happened so quickly after the invention of the electron tube. Similarly, random mutation does not seem to be sufficient to produce inventions like the airplane or the telephone in a reasonable amount of time. These more complicated inventions appear to call for a more sophisticated method of invention.

1.1 Invention via Heuristic Search

Several researchers, following in the footsteps of Newell and Simon [3], appear to have independently developed the idea that inventions could be realized through heuristic search. Weber and Perkins [4] adapted contemporary problem solving

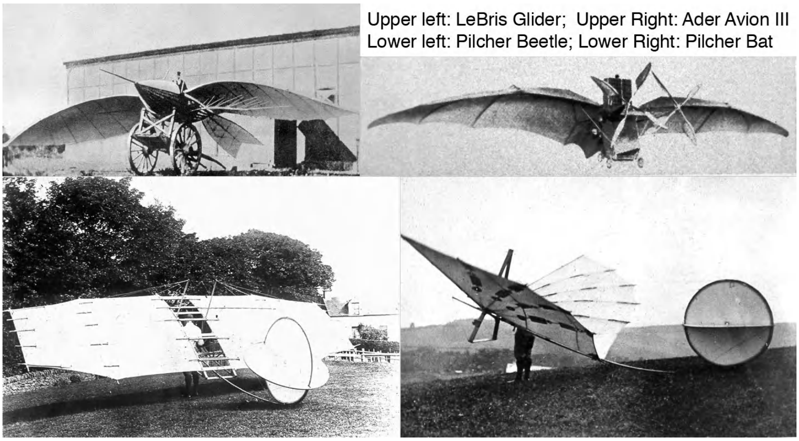

Upper left: LeBris Glider; Upper Right: Ader Avion III
Lower left: Pilcher Beetle; Lower Right: Pilcher Bat

Fig. 2. Designs created by *analogy to nature*. These inventors deliberately copied the structure of flying creatures in their designs. Although birds were a popular source of analogy, other flying creatures like bats and beetles were used as well.

to account for the process of invention. Following Simon's ([5], [6]) model of scientific discovery as heuristic search, Weber and Perkins described a set of heuristics to produce new inventions by performing a goal-directed search through a space of possible inventions based on a series of working-forward heuristics that enable a new set of inventions to be developed from existing ones. For example, the *join* heuristic creates a new invention by combining separate inventions together. The awl and the scraper can be combined using the join heuristic to produce a pointed knife. The knife, in turn, can be combined with a screwdriver, a pair of scissors, a corkscrew, and a saw into the contemporary Swiss army knife, a lightweight and versatile tool.

The join heuristic is a weak method, but is still far more powerful than random recombinations. Weber and Perkins restrict the join heuristic to combine functionally related objects. The heuristic would not be invoked to join a can opener with a computer monitor. Although such an implement would have greater functionality than the original inventions, the purposes of can openers and monitors have little to do with one another, so there is no reason to construct this awkward marriage.

In accounting for the invention of the airplane, Bradshaw & Lienert [7] and Bradshaw [8] also adopted a problem-solving perspective. Four design heuristics were identified from historical records of numerous different attempts to create a workable airplane from 1799 to 1909. Two of these heuristics, *analogy to nature* and *copycat* were the source of full designs, while two other heuristics, more is better and make small changes, were used to revise an existing design. Figure 2 illustrates different examples of the use of analogy, where various flying creatures (songbird, bat, and beetle) were used to inspire an airplane design. Figure 3 illustrates copycat, where a design produced by one inventor is adopted by another.

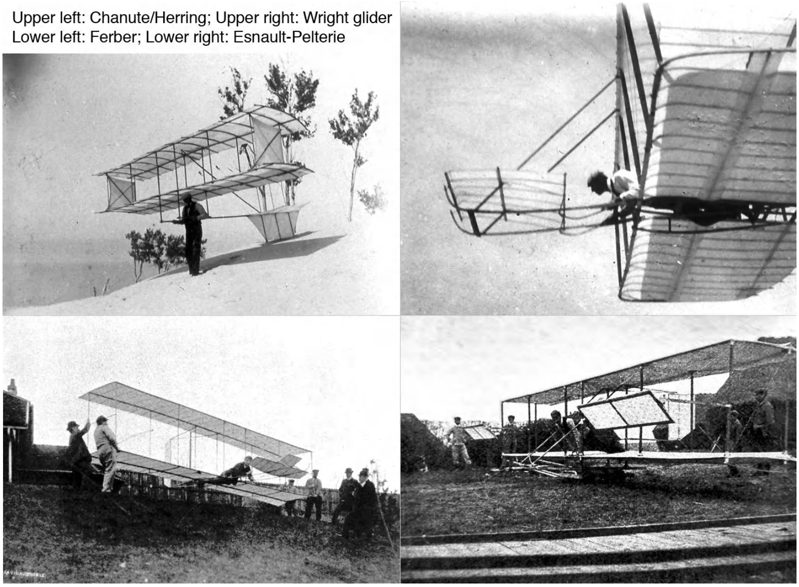

Upper left: Chanute/Herring; Upper right: Wright glider
Lower left: Ferber; Lower right: Esnault-Pelterie

Fig. 3. Airplanes designed by *copycat*. The Wrights copied the sturdy biplane design introduced by Octave Chanute and Augustus Herring, including the Pratt system of trussing the wings for strength. Ferber and Esnault-Pelterie both copied the Wright design. The failure of Esnault-Pelteries "perfect copy" of the Wright glider convinced Europeans that the Wrights were "bluffing" in their claims to have built airplanes.

Once an airplane had been designed, two additional heuristics could be used to revise the design. The first such heuristic is called more is better. Inventors added additional wings, propellers, tail surfaces, and other components to see if their design could be improved. Phillips, an English inventor, produced a design that had at least 196 different airfoils in four racks, not realizing that the turbulence produced by the forward wings would spoil the airflow over rearward wings. The second design-revision heuristic is known as *make small changes*. This covers a multitude of minor modifications made to an airplane, usually for ad hoc reasons.

Strategies for Searching Through the Design Space. These four heuristics alone can generate billions of different airplane designs given the parameters shown in Table 1. We will refer to the set of possible designs as the *design space* of the invention. Effective solutions (airplanes that are airworthy) are rare in this design space, so inventors need to find efficient means of searching through such a large space for rare solutions.

Most inventors relied upon a simple *design and test* strategy: They would design and build a craft, then take it out to the field to test it. Some craft were launched from a hill, others were launched from a rail system, while others attempted to fly from grassy meadows. Figure 6 illustrates the best performance

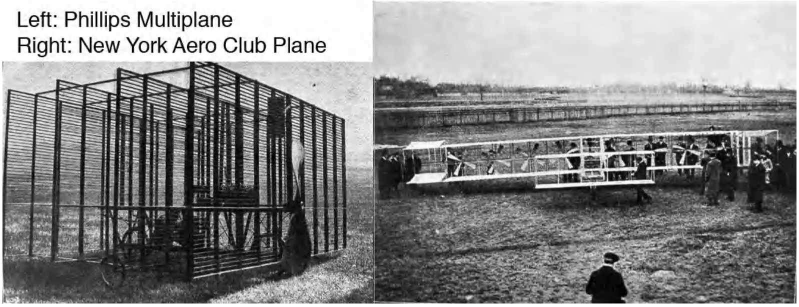


Fig. 4. Airplanes modified through the *more is better* heuristic. The Phillips Multiplane had approximately 196 different airfoils, while the New York Aero Club plane featured 8 propellers.

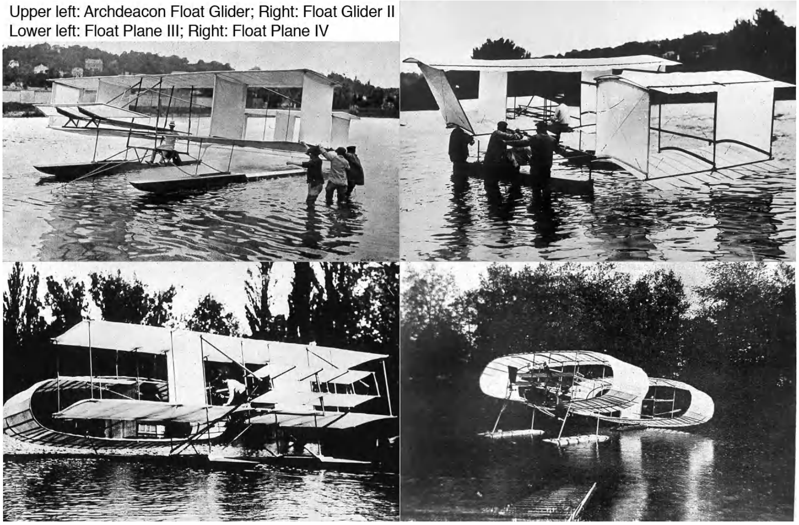

Upper left: Archdeacon Float Glider; Right: Float Glider II
Lower left: Float Plane III; Right: Float Plane IV

Fig. 5. Archdeacons airplane was modified several times *using make small changes.* The original design (a crude copy of the Wright craft) was changed by reducing the length of the lower wing, which led to the side curtains being placed at an angle. Then the rear wing was replaced with an ellipse (which suffers from particularly bad aerodynamic characteristics) and finally both wings were replaced by an ellipse.

of a number of craft between 1799 and 1909. The dotted line is the regression line for non-Wright craft in the period 1799 until December, 1905. In 1906 the Wrights patent first appeared, and so after that time inventors had access to information about an airworthy craft. The 1799-1905 regression line actually has a slightly negative slope, indicating that more capable craft were built near the beginning of this period than could be constructed at the end of the era: There was no substantial improvement in performance of airplanes over a 110-year

Table 1. Design Features of Para-Planes

Design Parameter	Possible Values
Number of wings.	1–196
Wing position	1–3 (monoplane, tandem, etc.)
Placement	stacked, tandem, staggered
Lateral arrangement	anhedral, flat, dihedral
Camber of wings	1–12, 1–6, etc.
Wingspan	6'–104'
Wing Chord	3'–10'
Shape of wings	bird-like, rectangular, bat-like, insect-like
Tail placement	forward (canard), rear, mid
Lateral control	none, wing warping, ailerons
Number of Propellers	0–8

span. The Wrights were a clear exception to this trend: their initial gliders were among the best-performing aircraft from the beginning, and they made steady progress. The 1903 craft (which was their first powered design) represents the transition between gliders and powered craft, and fits nicely on both regression functions.

The data shown in Figure 6 represents a considerable mystery: How were the Wrights able to develop an effective initial glider and sustain progress, while

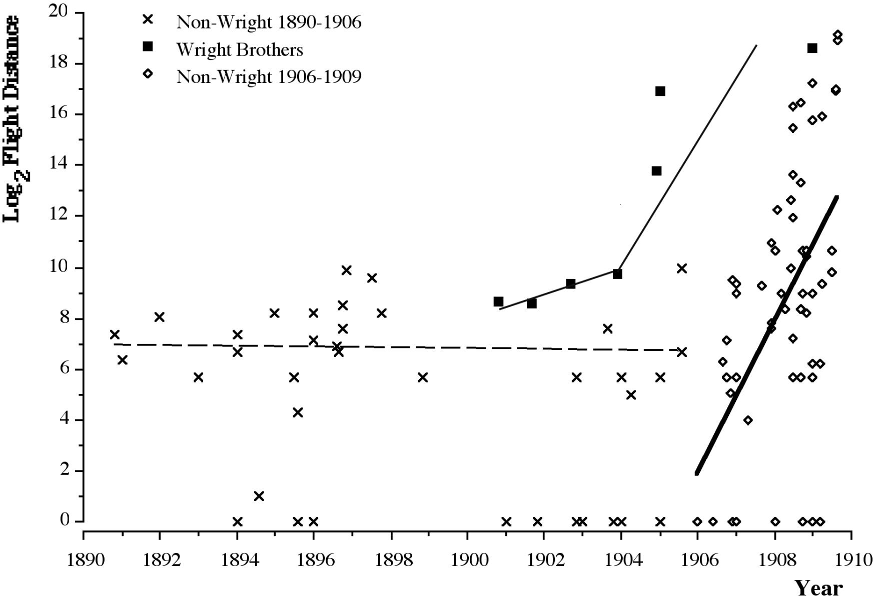


Fig. 6. Attempts to master the airplane. Each point represents the best flight made by a particular craft on the date the flight was made.

other inventors did not show similar improvements? To explain this mystery, we first need to examine the logic behind the design-and-test strategy of invention. Suppose that an inventor has designed an airplane, taken it out to the field, and flown it. Between 1799 and 1909, about the greatest distance achieved by any craft was 100 meters. Now the inventor would like to improve upon the design. As long as an inventor can create a craft that is even slightly better than the last, he can continue improvements and produce a positive performance slope. This represents a familiar hill-climbing strategy of solving problems, and hill-climbing is effective for solving many problems.

Why did this strategy fail in the invention of the airplane? Bradshaw & Lienert [7] argued that the failure arose because making a test flight does not provide diagnostic information about the strengths and weaknesses of a design: The wings may not have produced sufficient lift to keep the craft in flight, the airframe might be causing too much drag, the center-of-lift might not coincide with the center-of-balance, or the pilot may have made a mistake in flying the airplane. Under these circumstances, inventors had no reliable information about what was wrong with their craft, or what specifically to change to improve it.

So how did the Wrights escape this trap? Although they did build and test gliders and airplanes, their approach towards invention differed in substantial ways from their contemporaries. In particular they followed a *functional decomposition* strategy: They isolated different functional subsystems (wings, power plant, elevator), identified specific performance requirements for each of the systems (i.e., the wings must produce 110 kilos of lift), and employed what little was known about aerodynamics to produce a subsystem that met their design requirements. This led to the first glider, which they tested in 1900. When the glider did not perform as designed, the Wrights realized there was something wrong with their computations of lift. This led to their construction of a series of wind tunnels and the development of instruments to measure lift and drag. Somewhere between 80 and 200 different wing shapes were tested in the wind tunnel. Their results demonstrated that the current value of the *coefficient of lift* was incorrect, revealed a much better approximation for that coefficient, and showed that long and thin wings had better characteristics than short and broad ones. Wind tunnel tests also revealed that an airplane wing with its highest point near the leading edge of the wing had better changes in the travel of the center of lift than did a wing with the highest point in the center of the wing. Testing a functional subsystem in isolation produced diagnostic information about which wing designs were good, and which ones were bad. Their information was so precise that the Wrights were able to build a glider in 1902 that had excellent flight characteristics.

Between 1902 and 1903 the Wrights performed more tests in their wind tunnel – to determine the best shape for an airplane propeller. Once again their approach was quantitative: the Wrights knew how much horsepower their engine could produce, so they needed a propeller with a specific level of performance to produce the thrust their craft needed. The Wrights were so certain of the success of their 1903 flyer that the pair, who were known for their modesty and caution, wrote a press release announcing their success before leaving Dayton for Kitty Hawk. Their father issued the press release following the receipt of a telegram from the Wrights told of their actual success.

As an invention strategy, *functional decomposition* has several advantages over *design-and-test*. We have already discussed the lack of diagnosticity in design and test. Because of the decomposition inherent in functional decomposition, testing of a part is not confounded by the performance of other parts of the system. It was also possible for the Wrights to develop far more precise indices of performance of subsystems than it was to evaluate the performance of the part in a complete craft. In 1900 and 1901, for example, the Wrights suspected their wings were not producing as much lift as calculated. They attempted to measure the lift of their glider by weighting down the wings with some chain, flying their glider as a kite, and measuring the angle of attack and wind speed simultaneously. But the winds were not perfectly steady and their glider reacted by changing its elevation and angle of attack in reaction to changes in wind speed, so the Wrights could not determine with any degree of accuracy how much lift their wings were actually producing. They suspected that the value for the coefficient of lift was incorrect, but did not have proof. Once they built their wind tunnel, they were able to precisely measure lift, drag, and the coefficient of lift. A third advantage of functional decomposition arises from the combinatorics of *divide and conquer*: When one wing is shown to have poor lift and drag characteristics, the Wrights were able to exclude from consideration tens- or hundreds-of-thousands of airplanes: any design that included that wing was a bad choice.

Another advantage of the Wright approach arose from their use of precise performance specifications. The Wrights knew their gliders had to support the weight of the pilot along with the weight of the craft. For an 80-kilo pilot and a 45-kilo glider, the wings must produce 125 kilos of lift. Having these performance requirements allowed the Wrights to *satisfice* in their designs: once they found a wing that could produce the necessary lift at the target weight, they did not have to look further to find an optimal wing design. Also, having performance specifications allowed the Wrights to determine when their design had failed. The Wright's second glider, built in 1901, performed nearly as well as any other craft, powered or unpowered, had done to date. Rather than being satisfied with their near-world-record performance, the Wrights were discouraged that the craft had not performed as designed: this dissatisfaction led directly to their development of a wind tunnel to determine why the glider was not generating the computed lift.

One more aspect of the Wright's approach deserves mention: the use of theory and math to substitute for search. Previous research had uncovered a *lift function* that enabled the Wrights to test small models of wings just a couple of inches long, then predict the performance of a large-scale wing with considerable accuracy. If this function were not know, the Wrights might have been forced to test full-scale wings in a large wind tunnel. Such research would most likely have been prohibitive given their modest means.

Through all of these efficiencies, the Wrights were able to develop an airworthy glider in just three years, then take only three more to produce a practical airplane capable of extended flight. These accomplishments were made while they maintained a successful small business, and with quite modest financial resources. Only when others were able to study the Wright craft and the advances they made were they able to produce competitive airplanes.

1.2 Principles of Effective Invention

This review of the invention of the airplane suggests that there are ways to achieve considerable efficiency in the process of invention. These efficiencies will be most evident for complex inventions where various elements contribute to the success of the whole. Under such circumstances, search can be reduced by:

1. Identifying the functions to be performed by the invention;
2. Specifying functional requirements that the system must meet;
3. Developing subsystems that meet these functional requirements;
4. Testing the subsystems in isolation from the whole system;
5. Focusing attention on subsystems that fail to perform as designed; and
6. Utilizing theory to generalize results.

Whenever these strategies can be practically employed, they will reduce the complexity of the invention.

2 The Invention of AI Systems

In discussing invention, it should be clear that many AI methods, particularly those in learning and discovery systems, have application as a way to produce new inventions, just as they can learn and make new discoveries. Perhaps in the near future a discovery system will build a better mousetrap or a learning system will produce a better user interface. By exploring such applications we can increase the utility of our systems and methods: we can apply them to invention problems as well as discovery and learning problems. But there is another reason for discussing the process of invention: AI systems are not discovered they are invented. As such, they are governed by the same principles of invention that have just been described above.

Why are AI systems best considered as inventions and not discoveries? Clearly artificial intelligence falls within the“Sciences of the Artificial” as described by Herbert Simon [6]. AI may draw upon research findings in psychology, but clearly AI methods and systems are the product of human enterprise, and so can be understood as an invention. Given this status, we may consider how the lessons of invention can be applied to AI systems as a special case. We begin by considering the design space of AI systems. Table 2 illustrates some of the choices that investigators face as they are putting together a new AI system:

Many of the entries in the table refer to a family of related methods. For example, using parametric statistics to handle noise in the data might include

Table 2. Design Features of AI Systems

Design Parameter	Possible Values
Knowledge Representation	Symbols; Schemas; Propositions; Productions; Distributed Sub-Symbolic Nodes ...
Thought Processes	Productions; Bayesian Probabilities; Spreading Activation; Predicate Calculus; Schema Inference; Markov Transitions ...
Learning	Proceduralization; Composition; Backpropagation; Genetic Algorithms; ...
Noise	Parametric Statistics; Non-parametric statistics; Bayesian Probabilities; Signal Detection; ...
Test Database	Iris; Solar Flare; Credit Card; ...
Competing System	C4.5; Soar; ACT-R; Harmony; Neural Network; ...

something simple, like computing the mean, to finding a regression line, or even using the standard deviation to find outliers that are treated as a special case. Clearly we have produced a rich set of alternatives from which researchers can choose in developing a new AI system.

Let us suppose, for a moment, that a researcher decides to build a new AI system drawing upon the alternatives shown in Table 2. Choosing a system based upon schemas for knowledge representation, spreading activation and productions for thought processes, and proceduralization for learning, the researcher then adds a new method of dealing with noise based on non-parametric statistics and Bayesian probabilities, known as F.A.K.I.R., further adding to the pool of methods available in AI. The new system, including the F.A.K.I.R. algorithm, is called HOLIER.THAN.THOU.[1] The researcher decides to compare HOLIER.THAN.THOU. against C4.5 [9] using a database of credit card transactions.

On the first comparative test, HOLIER.THAN.THOU. does not perform well classifying database transactions. The researcher examines the errors made by the F.A.K.I.R. and identifies some problems with the new procedure for accommodating noise in the data. By adjusting several parameters and making other tweaks to the system, HOLIER.THAN.THOU. now outperforms C4.5 on the database by 2%.

We might ask, "What value is the new F.A.K.I.R. noise reduction technique for the AI community?" Clearly the researcher has demonstrated that, under some conditions, HOLIER.THAN.THOU. can out-perform C4.5. But grave questions remain about the generality of this claim, about the significance of the 2% difference, and about the source of the performance difference. Let us consider each of those issues in turn.

[1] Any resemblance between F.A.K.I.R/HOLIER.THAN.THOU. and an actual AI systems or methods is purely coincidental, and we do not suggest that any existing AI system has been developed in this way.

Generality of the Result. AI researchers often begin with an understanding of the problems that certain databases represent for learning and discovery algorithms. But we lack a deep conceptual understanding of the fundamentals: "How much adaptation does an adaptive system have to perform?" or "What kinds of learning does a learning system need to do?" We can answer these questions with respect to certain well-known databases, but not with respect to an entire class of problems. For this reason, we cannot readily determine how many different databases are needed to demonstrate the generality of a new system or algorithm. Should we test each new system on 10 different databases? Which ones? At what point are we certain that we have tested a new system against all of the interesting problems an intelligent system might face? The Wrights were lucky enough to have mathematical equations that allowed them to predict the performance of a full-size wing from a small model wing. Would anyone care to predict how HOLIER.THAN.THOU will do on a database of credit card transactions *vis a vis* a neural network system?

We may never enjoy the situation the Wrights found themselves in, where a simple mathematical function can predict how a system will behave under different situations. One way researchers have responded to questions about generalization is to test their system against multiple databases – a method that does help to establish the generality of a new method or system. Yet even still another serious issue remains: how good does our system or method need to be in order to be useful across an interesting range of problems? Remember that the Wrights could specify in advance how much lift their system needed to generate in order to fly. That allowed them to find a satisfactory solution to the problem, without the necessity of finding an optimal one. Are we now looking at solutions that are sub-satisfactory, satisfactory, approaching optimal, or optimal?

Significance of a Performance Difference. The researcher found a 2% performance advantage for HOLIER.THAN.THOU when compared to C4.5. But this advantage only occurred after careful fine tuning of HOLIER.THAN.THOU. Was the same care taken to ensure that C4.5 was performing at its best? Perhaps, perhaps not. Even if C4.5 was adjusted to perform at its best, we are still uncertain about what the best possible performance is on the credit card database. It might seem possible to attain a 100% accuracy on classifying all items in the database simply by memorizing each item. However, some databases could suffer from an impoverished description of items where two items with the same description belong to different classes, or God could be playing dice with the Universe, and no description would be adequate to classify every item correctly.

A more difficult question arises when we try to determine whether the 2% improvement in accuracy rate is *better* or *worse.* At first glance the question seems foolish: HOLIER.THAN.THOU performed more accurately on the database. How could that not be better? Yet experience has taught us that adaptive systems can learn the noise in the database as well as the signal. These systems do not generalize as well to new data as ones that only learn the true generalizations present in the database.

A final awkward question arises when we consider the statistical and practical significance of a 2% improvement in classification performance. If C4.5 correctly classifies 97% of the transactions and HOLIER.THAN.THOU correctly classifies 99%, the difference could be significant and important. But if C4.5 correctly classifies 20% of the transactions and HOLIER.THAN.THOU classifies 22%, neither system seems very impressive. Researchers in AI commonly use some sort of split-part reliability computation, which helps to determine the reliability and statistical significance of the results.

Credit Assignment for F.A.K.I.R and HOLIER.THAN.THOU. Once we convince ourselves that the 2% difference is significant and important, we are left with one more awkward question: where did this difference come from? Was it due to some advantage of the F.A.K.I.R. algorithm in isolating noise and concept drift from the signal? Or was it due to a difference in the initial representation of the data between HOLIER.THAN.THOU and C4.5? Or was there some other difference between the two systems? Answering the question has greater importance than at first appears. Perhaps the F.A.K.I.R. algorithm is improving the performance of HOLIER.THAN.THOU by 25%, but other limitations of HOLIER.THAN.THOU reduce the advantage by 23%. We might then combine F.A.K.I.R. with C4.5 and achieve a better result yet. By knowing how well each element of the system is doing its job, we can produce the best possible combination of elements.

The analogy between the invention of the airplane and the invention of AI systems can be pushed too far: Computer programs are typically designed through a series of function calls, and it is possible to determine if the calls are operating as designed. This lends a transparency to computer programs that airplane inventors did not enjoy. Function calls often map roughly onto the functional specifications for a system, although there are many internal function calls that do not have an obvious connection to the larger functional subsystems of the program. Yet by considering AI as an invention, it raises two important questions: “What are we trying to invent?” and “Are we working efficiently toward that goal?”

2.1 Reflections on Invention and AI

When one examines recent developments in AI, it is clear that AI researchers are inventive and are pouring tremendous creative energy into developing new heuristic and algorithmic methods to address difficult problems. Evidence of this inventiveness is present in the the two volumes published last year for ALT 04 [10] and Discovery Sciences 04 [11], each of which presented a number of important papers in their respective fields. As a result of this worldwide enterprise, researchers are now faced with an embarrassment of riches in the number of different methods they have available for the construction of new AI systems.

But there still seems to be an important gap in our knowledge – we don’t fully understand the relationship between where we are and where we want to be. Are we building AI systems like the pre-Wright airplanes that struggled to 'fly' 100 meters? Or are we improving capable airplanes to extend their range from trans-continental flight to inter-continental flight? With aviation, everyone knew that airplanes needed to fly long distances quickly and to carry as much weight as possible. AI has no such simple goals: it may be quite valuable to build an expert system like XCON [12] to design the backplane of Vax computers, even if the system has only a limited expertise and lifetime. Or we may set our goals higher to develop more versatile and capable AI systems.

We can, of course, continue to develop even more new learning and discovery methods. But hopefully the energy being spent to develop new methods can be balanced with an effort to better document *what our systems need to do* and *how well they need to do those things.* As the baseball player Yogi Berra said, "If you don't know where you are going, you will wind up somewhere else." Through a better understanding of the fundamental problems of learning, discovery, and AI, we can work towards functional specifications that tell us how well our systems need to perform, then choose methods that will get us where we want to be, instead of somewhere else.

References

[1] Basalla, G.: The Evolution of Technology. Cambridge University Press (1988)

[2] Petroski, H.: The Evolution of Useful Things. Alfred A. Knopf. (1993).

[3] Newell, A., & Simon, H.A.: Human Problem Solving Prentice-Hall (1972)

[4] Weber, R.J. & Perkins, D.N.: How to invent artifacts and ideas. New Ideas in Psychology, **7** (1989) 49–72.

[5] Simon, H.A.: Scientific discovery and the psychology of problem solving. In R. Colodny (Ed.), Mind and Cosmos. (1966) Pittsburgh: University of Pittsburgh Press.

[6] Simon, H.A.: The Sciences of the Artificial, Second Edition. The MIT Press. (1981)

[7] Bradshaw, G.L., & Lienert, M.: The Invention of the Airplane Proceedings of the Thirteenth Annual Cognitive Science Conference (1991), pp. 605-610.

[8] Bradshaw, G.L.: The Airplane and the Logic of Invention. In R.N. Giere (Ed.), Cognitive Models of Science. The University of Minnesota Press. (1992)

[9] Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kauffman. (1993)

[10] Ben David, S, Case, J., & Maruoka, A.: Algorithmic Learning Theory 15th International Conference, ALT 2004, Padova, Italy. Springer-Verlag (2004)

[11] Suzuki, E., & Arikawa, S.: Discovery Science 7th International Conference, DS 2004, Padova, Italy. Springer-Verlag (2004)

[12] McDermott, J.: R1: A Rule-based Configurer of Computer Systems. Artificial Intelligence, **19** (1982), 39–88

Algorithms and Software for Collaborative Discovery from Autonomous, Semantically Heterogeneous, Distributed Information Sources[*]

Doina Caragea, Jun Zhang, Jie Bao, Jyotishman Pathak, and Vasant Honavar

Artificial Intelligence Research Laboratory,
Center for Computational Intelligence, Learning, and Discovery,
Department of Computer Science, Iowa State University,
226 Atanasoff Hall, Ames, IA 50011
honavar@cs.iastate.edu

Abstract. Development of high throughput data acquisition technologies, together with advances in computing, and communications have resulted in an explosive growth in the number, size, and diversity of potentially useful information sources. This has resulted in unprecedented opportunities in data-driven knowledge acquisition and decision-making in a number of emerging increasingly data-rich application domains such as bioinformatics, environmental informatics, enterprise informatics, and social informatics (among others). However, the massive size, semantic heterogeneity, autonomy, and distributed nature of the data repositories present significant hurdles in acquiring useful knowledge from the available data. This paper introduces some of the algorithmic and statistical problems that arise in such a setting, describes algorithms for learning classifiers from distributed data that offer rigorous performance guarantees (relative to their centralized or batch counterparts). It also describes how this approach can be extended to work with autonomous, and hence, inevitably semantically heterogeneous data sources, by making explicit, the ontologies (attributes and relationships between attributes) associated with the data sources and reconciling the semantic differences among the data sources from a user's point of view. This allows user or context-dependent exploration of semantically heterogeneous data sources. The resulting algorithms have been implemented in INDUS - an open source software package for collaborative discovery from autonomous, semantically heterogeneous, distributed data sources.

[*] The full version of this paper is published in the Proceedings of Algorithmic Learning Theory, the 16th International Conference, ALT 2005, Lecture Notes in Artificial Intelligence Vol. 3734.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, p. 14, 2005.

Training Support Vector Machines via SMO-Type Decomposition Methods*

Pai-Hsuen Chen, Rong-En Fan, and Chih-Jen Lin

Department of Computer Science,
National Taiwan University

Abstract. This article gives a comprehensive study on SMO-type (Sequential Minimal Optimization) decomposition methods for training support vector machines. We propose a general and flexible selection of the two-element working set. Main theoretical results include 1) a simple asymptotic convergence proof, 2) a useful explanation of the shrinking and caching techniques, and 3) the linear convergence of this method. This analysis applies to any SMO-type implementation whose selection falls into the proposed framework.

[*] The full version of this paper is published in the Proceedings of Algorithmic Learning Theory, the 16th International Conference, ALT 2005, Lecture Notes in Artificial Intelligence Vol. 3734.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, p. 15, 2005.

The Robot Scientist Project

Ross D. King, Michael Young, Amanda J. Clare, Kenneth E. Whelan,
and Jem Rowland

The University of Wales, Aberystwyth
{rdk, miy, afc, knw, jjr}@aber

Abstract. We are interested in the automation of science for both philosophical and technological reasons. To this end we have built the first automated system that is capable of automatically: originating hypotheses to explain data, devising experiments to test these hypotheses, physically running these experiments using a laboratory robot, interpreting the results, and then repeat the cycle. We call such automated systems "Robot Scientists". We applied our first Robot Scientist to predicting the function of genes in a well-understood part of the metabolism of the yeast *S. cerevisiae*. For background knowledge, we built a logical model of metabolism in Prolog. The experiments consisted of growing mutant yeast strains with known genes knocked out on specified growth media. The results of these experiments allowed the Robot Scientist to test hypotheses it had abductively inferred from the logical model. In empirical tests, the Robot Scientist experiment selection methodology outperformed both randomly selecting experiments, and a greedy strategy of always choosing the experiment of lowest cost; it was also as good as the best humans tested at the task. To extend this proof of principle result to the discovery of novel knowledge we require new hardware that is fully automated, a model of all of the known metabolism of yeast, and an efficient way of inferring probable hypotheses. We have made progress in all of these areas, and we are currently 6building a new Robot Scientist that we hope will be able to automatically discover new biological knowledge.

1 Introduction

1.1 The Robot Scientist Concept

The Robot Scientist project aims to develop computer systems that are capable of automatically: originating hypotheses to explain data, devising experiments to test these hypotheses, physically running these experiments using a laboratory robot, interpreting the results, and then repeat the cycle (Figure 1).

1.2 Motivation

- Philosophical - Our primary motivation is a better understanding of science. For us, the question of whether it is possible to automate the scientific discovery process is central to an understanding science, as we believe that we do not fully understand a phenomenon unless we can make a machine, which reproduces it.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 16 – 25, 2005.

- Technical - In many areas of science our ability to generate data is outstripping our ability to analyse the data. One scientific area where this is true is post-genomic Biology where data is now being generated on an industrial scale. We contend that the analysis of scientific data needs to become as industrialized as its generation.

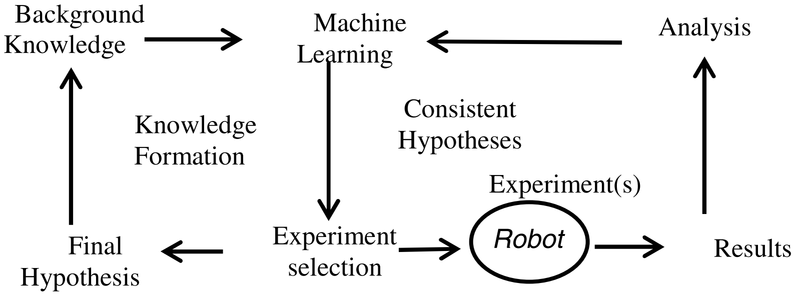


Fig. 1. The Robot Scientist Hypothesis Generation, Experimentation, and Knowledge Formation loops

1.3 Scientific Discovery

The branch of Artificial Intelligence devoted to developing algorithms for acquiring scientific knowledge is known as “scientific discovery”. The pioneering work in the field was the development of learning algorithms for analysis of mass-spectrometric data [1]. This work was notable as an early example of interdisciplinary research: it involved world-class scientists from biology (J. Lederberg), chemistry (C. Djerassi), and computer science (E. Feigenbaum). This project initiated the whole field of machine learning. In the subsequent 30 years, much has been achieved, and there are now a number of convincing examples where computer programs have made explicit contributions to scientific knowledge [2,3]. However, the general impact of such programs on science has been limited. This is now changing, as the confluence of the expansion of automation in science, and advances in AI, are making it increasingly possible to couple scientific discovery software with laboratory instrumentation.

2 Previous Work on the Robot Scientist

In [4] we first developed the Robot Scientist concept. The Robot Scientist is a reasoned, but radically new, approach to scientific discovery that seeks to integrate data generation and analysis in a physically closed loop. A widely accepted view of science is that it follows a “hypothetico-deductive” process [5]. Scientific expertise and imagination are first used to form possible hypotheses, and then the deductive consequences of these hypotheses are tested by experiment. The Robot Scientist methodology (Figure 1) is consistent with this paradigm: we employ the logical inference mechanism of abduction [6] to form new hypotheses, and that of deduction to test which hypotheses are consistent.

2.1 The Biological System

The first aim of the first Robot Scientist project was to develop a proof-of-principle system that would demonstrate automated cycles of hypothesis generation and experiment on a real biological system. For this we chose the scientific area of "Functional Genomics". The aim of this branch of biology is to both uncover the function of genes identified from sequencing projects (such as that on the human genome), and to better characterize the function of genes with currently putative functions. We chose to focus on brewer's (or baker's) yeast (*S. cerivisae*). This is the best understood eukaryotic organism. As humans are eukaryotic organisms, this yeast is used as a "model" of human cells, as it is simpler and easier to work with. *S. cerivisae* was the first eukaryotic organism sequenced and has been studied for over a hundred and fifty years. Despite this, around 30% of its ~6,000 genes still have no known function.

A key advantage with working with yeast is that it is possible to obtain strains of yeast with each of the ~6,000 genes knocked out (removed). We chose to use these mutatnts along with a classical genetic technique known as "auxotrophic growth experiments". These experiments consist of making particular growth media and testing if the mutants can grow (add metabolites to a basic defined medium). A mutant is auxotrophic if cannot grow on a defined medium that the wild type can grow on. By observing the pattern of metabolites that recover growth, the function of the knocked out mutant can be inferred. We focused on the aromatic amino acid (AAA) pathway in yeast.

2.2 Logical Model

In any scientific discovery problem that is not purely phenomenological we need to develop a model of the natural system. We therefore developed a logical formalism for modelling cellular metabolism that captures the key relationships between protein-coding sequences (genes, ORFs), enzymes, and metabolites in a pathway, along with feedback, etc. [6]. This model is expressed in predicate logic and encoded in the logic programming language Prolog (see Figure 2.).

Logic is our oldest (>2,500 years) and best understood way of expressing knowledge, and computer programs are the most general way we have of expressing knowledge: logic programs combine the clarity of logic with the expressive power of computer programs. All objects (genes, proteins, metabolites) and their relationships (coding, reactions, transport, feed-back) are described as logical formulae. The structure of the metabolic models pathway is that of directed graphs, with metabolites as nodes and enzymes as arcs. An edge arc corresponds to a reaction. The compounds at each vertex node are the set of all metabolites and the compounds that can be synthesised by the reactions leading to it. Reactions are modelled as unidirectional transformations. A model's consistency and completeness can be analysed by comparing the model's logical consequences with the outcomes of *in vivo* auxotrophic growth experiments. The model can thus be used to yield a procedural specification of the functional genomics problem, namely how to infer gene functions from experimental observations. The model is both declarative (expressing text-book biochemistry) and procedural (enabling inferences about pathways). In particular, two types of inference can be made: deductions to infer phenotype, and abductions to infer missing reactions (gene functions) see Figure 3. A mutant is inferred (deduced) to grow if and only if, a path can be found from the input metabolites to the three aromatic amino acids. Conversely, a mutant is inferred to be auxotrophic if, and only if, no such path can be found. We formed our "gold-standard" AAA model to fit both the existing knowledge on the AAA pathway and our experimental auxotrophic growth experiments.

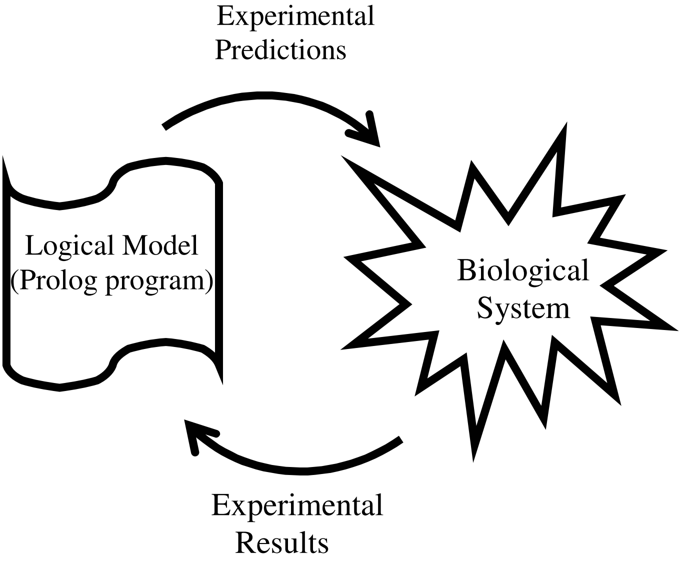


Fig. 2. The Relationship between the logical model and the experimental System

Deduction

Rule: If a cell grows, then it can synthesise tryptophan.
Fact: cell cannot synthesise tryptophan
∴ Cell cannot grow.

Given the rule $P \rightarrow Q$, and the fact $\neg Q$, infer the fact $\neg P$ (*modus tollens*)

Abduction

Rule: If a cell grows, then it can synthesise tryptophan.
Fact: Cell cannot grow.
∴ Cell cannot synthesise tryptophan.

Given the rule $P \rightarrow Q$, and the fact $\neg P$, infer the fact $\neg Q$

Fig. 3. Simplified form of the deductive and abductive inference used

The form of the hypotheses that were abductively inferred was very simple. Each hypothesis binds a particular gene to an enzyme that catalyses the reaction. For example:

- A correct hypothesis would be that: YDR060C codes for the enzyme for the reaction: chorismate → prephenate.
- An incorrect hypothesis would be that: it coded for the reaction: chorismate → anthranilate.

2.3 Active Learning

The branch of machine learning that deals with algorithms that can choose their own examples (experiments) is known as "active learning" [7]. If we assume that each hypothesis has a prior probability of being correct, and that each experiment has an associated price, then scientific experiment selection can be formalised as the task of: given a set of possible hypotheses, each with a probability of being correct, and given that each experiment has an associated cost, select the optimal series of experiments (in terms of expected cost) to eliminate all but the one correct hypothesis [8]. This problem is, in general, computationally intractable (NP-hard). However, it can be shown that the experiment selection problem is structurally identical to finding the smallest decision tree, where experiments are nodes, and hypotheses leaves. This is significant because a Bayesian analysis of decision-tree learning has shown that near-optimal solutions can be found in polynomial time [9]. To approximate the full Bayesian solution we use the following [8]. *EC(H,T)* denote the minimum expected cost of experimentation given the set of candidate hypotheses *H* and the set of candidate trials *T*:

$$EC(\varnothing, T) = 0$$

$$EC(\{h\}, T) = 0$$

$$EC(H,T) \approx \min_{t \in T}\left[C_t + p(t)(mean_{t' \in (T-t)} C_{t'})J_{H[t]} + (1 - p(t))mean_{t' \in (T-t)} C_{t'})J_{H[\bar{t}]}\right]$$

$$J_H = -\sum\nolimits_{h \in H} p(h) \lfloor \log_2(p(h)) \rfloor$$

Ct is the monetary price of the trial *t*

p(t) is the probability that the outcome of the trial *t* is positive

p(t) can be computed as the sum of the probabilities of the hypotheses (*h*) which are consistent with a positive outcome of *t*.

2.4 Results

A Robot Scientist was physically implemented that can conduct biological assays with minimal human intervention after the robot is set up [4]. The hardware platform consisted of a liquid-handling robot (Biomek 2000) with its control PC, a plate reader (Wallac 1420 Multilabel counter) with its control PC, and a master PC to control the system and do the scientific reasoning. The software platform consisted of background knowledge about the biological problem, a logical inference engine, hypothesis generation code (abduction), experiment selection code (deduction), and the Laboratory Information Management System (LIMS) code that glued the whole system together. We used the Inductive Logic Programming (ILP system ASE-Progol. The robot conducted experiments by pipetting and mixing liquids on microtitre plates. Given a computed definition of one or more experiments, we developed code which designed a layout of reagents on the liquid-handling platform that would enable these experiments, with controls, to be carried out efficiently. In addition, the liquid-handling robot was automatically programmed to plate out the yeast and media into the correctly specified wells. The system measured the concentration of yeast in the wells of the microtitre trays using the adjacent plate reader and returns the results to the LIMS (although microtitre trays were still moved in and out of incubators manually). *The key point is that there was no human intellectual input in the design of experiments or the interpretation of data.*

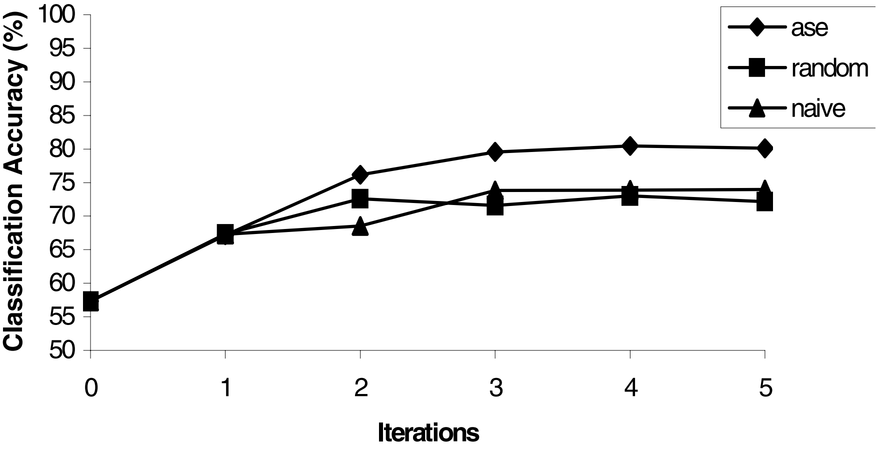


Fig. 4. The observed classification accuracy versus iterations (time)

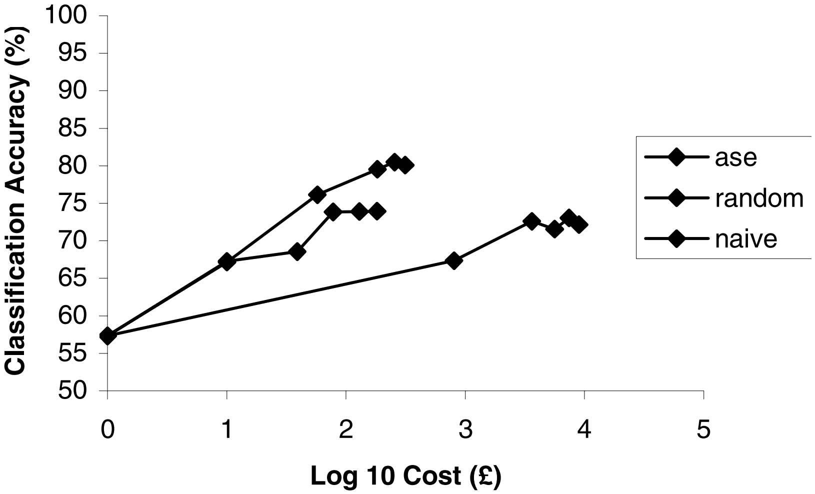


Fig. 5. The Observed classification accuracy versus cost (price of chemicals)

Figures 4 (below) shows the average classification accuracy versus experimental iteration (time) for the robot's intelligent strategy (red) compared with random (blue), and the naïve strategy of always choose the cheapest experiment (green). Figure 5 shows the average classification accuracy versus money spent (£). The intelligent strategy is both significantly faster and cheaper than the other strategies. When compared with human performance on this task, the Robot was as good as the best human scientists.

3 Current Status of the Robot Scientist

In our first work with the Robot Scientist we demonsrtaed a proof-of-principle application of the Robot Scientist. We demonsrtaed that we could automatically *rediscover* known biological knowledge. We now wish to extend this result to the *discovery* of new biological knowledge. To achieve this we have chosen to focus on the same biological problem. However to actually fully automattedly discover new knowledge a number of extensions are required:

- New Hardware
  - The original hardware was not fully automated, and several steps had to be done manually at the request of the Robot Scientist. We wish to make the system fully automated.
  - The experimental throughput capacity of the original hardware was also limited. A key advantage of automation is that it can be scaled up. The new hardware will have far greater capacity.
  - We will also extend the original qualitative experimental methodology (growth v no-growth) to a quantitative measurement of growth.
- Expansion of the background knowledge to include as much as possible of what is known about yeast metabolism. For if the Robot Scientist does not already know what is scientifically known, then it will be very difficult to discover something novel. This will require a move from a model with ~10 reactions to a model with more than 1,000 reactions. Our current model includes 1,166 genes (940 known, 226 inferred). As in the original AAA model, growth is predicted if there exists a path from the growth medium to defined end-points.
- Improve the efficiency of the hypothesis generation method. The current approach is purely logical and does not take advantage of domain knowledge. This approach will not scale to a model of two orders of magnitude greater size. Therefore, we will use bioinformatics to incorporate biological knowledge. One way of thinking about current bioinformatic genome annotation is as *hypothesis formation processes*; and hypothesis formation is perhaps the hardest part of automating science. Therefore, bioinformatic methods will generate the hypotheses that the robot scientist will experimentally test.

3.1 The New Robot Scientist Hardware

Our new Robotic Scientist hardware will be commissioned in the last quarter of 2005, and will cost £450,000 (see Figure 6). It will be manufactured by Caliper Life

Sciences. The hardware will consist of the following components: -80C freezer, a liquid-handling robot, incubator(s), plate-reader(s), and robot arms. The robotic system is designed to be able to be able to in a completely automatic manner: select frozen yeast strains from the freezer, inoculate these strains into a rich medium, harvest a defined quantity of cells, inoculate these cells into specified media (base plus added metabolites and/or inhibitors), and accurately measure growth in the specified media. Design of this system has been extremely challenging, and the specification has taken over 6 months to refine and make practical. To the best of our knowledge, after extensive discussions with manufacturers, we are confident that there is no comparable system anywhere in the world that can flexibly automate anything close to as many growth experiments. The system will be capable of initiating >1,000 new strain/defined growth-medium experiments a day, and each experiment will last up to 3days (plus an initiation day), using a minimum of 50 different yeast strains. It will be possible to take an optical density (OD) measurement for an experiment every 20 minutes, enabling accurate growth curves to be formed. It will also be possible to take samples from experiments for more detailed analysis, or to inoculate other experiments. The system will be able to run "lights out" for days at a time.

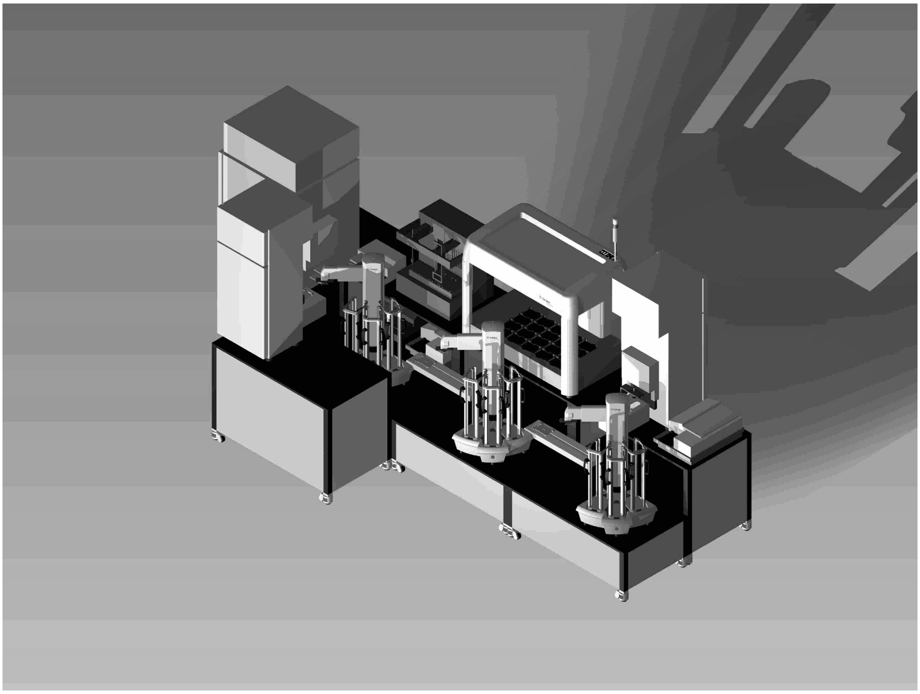


Fig. 6. Sketch of the new Robot Scientist hardware

The class of the experiments possible using this new hardware is comparable to those that the Robot Scientist can currently undertake. However the major advances will be:

- A huge increase in the scale of the number of experiments performed. Using our existing robotic system, we can perform ~200 strain/medium growth measurements a day: with our new robotic system we will be able to perform >100,000.
- A reduction in experimental noise. The current laboratory robot has ~25% noise when assaying growth or no-growth - mostly due to it being in a non-sterile environment, and cross-plate contamination. This noise will be drastically reduced, increasing throughput (through fewer controls being required), and simplifying data analysis.
- Accurate quantitative measurement of growth. Most genes display quantitative rather than qualitative effects under most environmental conditions [*Ol2*].
- Measure growth curves and yield.
- An increase in the range of metabolites used. We plan to have ~500 metabolites available, compared to a ~50 at present.
- The use of specific enzyme inhibitors.
- An increase in the range of strains used: including a set of Canadian double knockouts, and a set of knock-down mutants: where essential genes have been placed under the control of a promoter (e.g. *tet*O).
- The experiments will be fully automatic. Currently the Robot Scientist needs to direct a technician to execute a number of steps.

3.2 The Experimental Plan

The plan is to initiate ~1,000 experiments a day, providing ~200,000 daily measurements (based on a 3-day cycle measuring every 20 mins.). The reason that so many experiments are required is that even relatively simple cells, such as those of *S. cerevisiae*, are extremely complicated systems involving thousands of genes, proteins, and metabolites. Such systems can be in astronomical numbers of states, and the only possible way to dissect them is to be intelligent, and to do large numbers of experiments. One way to think about it is as an information theory problem: a complicated message cannot be sent using a few bits. Note, we do not plan to do all possible experiments, as even to test all possible pairs of metabolites would involve: 6,000 (genes) * (500 (metabolites))^2 = 1,500,000,000 (experiments).

All results will be stored in the Bio-Logical relational database (see above) along with meta-data detailing the experimental conditions. We expect to produce >40,000,000 growth measurements and all these results will be placed in the public domain. On their own, these results will constitute a significant contribution to scientific knowledge. N.B. the existing bioinformatic information on the growth of knockouts is often very poor, i.e. often gene knock-outs labelled as “lethal” have no description of the growth medium used, and the information is often also unreliable as it was produced using noisy high-throughput screens.

4 Discussion

The general Robot Scientist idea could be applied to many scientific problems. We are actively investigating the following two areas:

- Drug design - selection of compounds from libraries and/or use of laboratory on chip technology. The idea here is to incorporate the Robot Scientist into a Quantitative Structure Activity Relationship (QSAR) system [10].
- Quantum control - using femtosecond (10^{-15}s) lasers to control chemical synthesis. We are collaborating with the Department of Chemistry at the University of Leeds (UK) to use Robot Scientist type ideas to control the search for patterns of femtosecond laser pulses that can act as “optical catalysts” [11]. The main difference with this application and those in yeast is that the experiments take ~1s (and could be as low as 0.001s), compared to 24hours with yeast.

Acknowledgements

We are grateful for support from the BBSRC (2/E11552, E20268) and HEFCW (Aber 4). Mike Benway of Claiper Life Sciences made an important contribution to the new the system design.

References

1. Buchanan, B.G., Sutherland, G.L., Feigenbaum, E.A. Toward automated discovery in the biological sciences. In Machine Intelligence 4, Edinburgh University Press, (1969) 209-254
2. Langley, P., Simon, H.A., Bradshaw, G.L., Zytkow, J.M.: Scientific Discovery: Computational Explorations of the Creative Process. MIT Press (1987)
3. King, R.D., Muggleton, S.H., Srinivasan, A., Sternberg, M.J.E.: Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. Proc. Nat. Acad. Sci. USA 93, (1996) 438-442
4. King R.D, Whelan K.E, Jones F.M, Reiser P,J,K Bryant C.H, Muggleton S, Kell D.B, Oliver S.: Functional Genomic Hypothesis Generation and Experimentation by a Robot Scientist. Nature 427 (1994) 247-252
5. Popper, K. The Logic of Scientific Discovery, Hutchinson, London (1972)
6. Reiser, P.G.K., King, R.D., Kell, D.B., Muggleton, S.H., Bryant, C.H. Oliver, S.G.: Developing a logical model of yeast metabolism. Electronic Transactions in Artificial Intelligence 5, (2001) 223-244
7. Duda, R.O., Hart, P.E., Stork, D.G. Pattern Classification. Wiley. (2001)
8. Bryant, C.H., Muggleton, S.H., Oliver, S.G., Kell, D.B., Reiser, P, King, R.D.: Combining inductive logic programming, active learning, and robotics to discover the function of genes. Electronic Transactions in Artificial Intelligence. 5 (2001) 1-36
9. Muggleton, S. Page, D.: A learnability model of universal representations and its application to top-down induction of decision trees. In K. Furukawa, D. Michie, & S. Muggleton (eds.) Machine Intelligence 15, Oxford University Press, (1999) 248-267.
10. King, R.D., Muggleton, S., Lewis R.A., Sternberg, M.J.E.: Drug design by machine learning: The use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. Proc. Nat. Acad. Sci. U.S.A. 89. (1992) 11322-11326.
11. Levis, R.J., Menkir, G., Rabitz H.: Selective Covalent Bond Dissociation and Rearrangement by Closed-Loop, Optimal Control of Tailored, Strong Field Laser Pulses, Science, 292, (2001)709-713

The Arrowsmith Project: 2005 Status Report

Neil R. Smalheiser

UIC Psychiatric Institute, University of Illinois-Chicago, MC912,
1601 W. Taylor Street, Chicago, IL 60612 USA
neils@uic.edu

Abstract. In the 1980s, Don Swanson proposed the concept of "undiscovered public knowledge," and published several examples in which two disparate literatures (i.e., sets of articles having no papers in common, no authors in common, and few cross-citations) nevertheless held complementary pieces of knowledge that, when brought together, made compelling and testable predictions about potential therapies for human disorders. In the 1990s, Don and I published more predictions together and created a computer-assisted search strategy ("Arrowsmith"). At first, the so-called one-node search was emphasized, in which one begins with a single literature (e.g., that dealing with a disease) and searches for a second unknown literature having complementary knowledge (e.g. that dealing with potential therapies). However, we soon realized that the two-node search is better aligned to the information practices of most biomedical investigators: in this case, the user chooses two literatures and then seeks to identify meaningful links between them. Could typical biomedical investigators learn to carry out Arrowsmith analyses? Would they find routine occasions for using such a sophisticated tool? Would they uncover significant links that affect their experiments? Four years ago, we initiated a project to answer these questions, working with several neuroscience field testers. Initially we expected that investigators would spend several days learning how to carry out searches, and would spend several days analyzing each search. Instead, we completely re-designed the user interface, the back-end databases, and the methods of processing linking terms, so that investigators could use Arrowsmith without any tutorial at all, and requiring only minutes to carry out a search. The Arrowsmith Project now hosts a suite of free, public tools. It has launched new research spanning medical informatics, genomics and social informatics, and has, indeed, assisted investigators in formulating new experiments, with direct impact on basic science and neurological diseases.

1 Introduction

In the 1980s, Don Swanson proposed the concept of "undiscovered public knowledge." He published several examples [1-4] in which two disparate literatures (i.e., sets of articles having no papers in common, no authors in common, and few cross-citations) nevertheless held complementary pieces of knowledge that, when brought together, made compelling and testable predictions about potential therapies for human disorders. I was conducting neuroscience research and teaching a course on "The Process of Scientific Discovery" at University of Chicago in the early 1990s

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 26 – 43, 2005.

when I got a phone call from Don, asking me if I could explain an apparent artifact in his recent analysis of “magnesium” as a term that was pervasive in the neuroscience literature. Everywhere he looked, no matter what neurological disease, it seemed that magnesium was implicated! I explained that this was no artifact – indeed, glutamate excitotoxicity and flow of calcium ions through the NMDA glutamate receptor were thought to be fundamentally important in neurological pathophysiology in a wide variety of conditions, and magnesium was an endogenous factor that controlled the permeability of this receptor to calcium ions. This brief phone call led to a collaboration that now has stretched well over a decade. In the ‘90s, we published a number of literature-based predictions together [5-8] and created a computer-assisted search strategy (“Arrowsmith”) for literature-based discovery [9, 10]. Don then created a free, public demonstration website for conducting Arrowsmith searches, though perhaps the turning point in evolution of the project occurred when Ron Kostoff of the Office of Naval Research asked us to conduct a one-year pilot study to test whether Arrowsmith searches could be used to assist intelligence officers in gathering and integrating disparate pieces of information [11, 12].

These experiences suggested that Arrowsmith might be ready for testing among a wider audience of investigators. Since I am a neuroscientist, it was natural to focus on the biomedical community, but at the same time, it was not clear whether most bench scientists wanted or needed the kind of information that Arrowsmith could provide. Would they find routine occasions for using such a sophisticated tool? Furthermore, as of 2001, it took many hours to carry out a single search, including crafting Arrowsmith queries, navigating the website, and analyzing the results. Would typical biomedical investigators be sufficiently motivated to learn to carry out Arrowsmith analyses? Would they uncover significant findings that affect their experiments or suggest new research directions? Even more uncertain was the question of which, if any, funding agency would support development and testing of Arrowsmith. Fortunately, Stephen Koslow of NIMH had spearheaded a unique NIH-wide program called the Human Brain Project, which sought to establish an informatics infrastructure for neuroscientists to pursue a new paradigm of scientific investigation – one which does not simply formulate self-contained hypotheses, but integrates concepts across disciplines and across investigators. His philosophy was that it is not enough to acquire new data; rather, scientists must be able to carry out data mining, data sharing and data re-use [i.e. re-analyze previous experiments done by others]. The Human Brain Project issued grants that were also unique – each funded program had to combine neuroscience research and informatics research – and they showed enthusiastic support for the Arrowsmith project.

Our early publications had emphasized the so-called one-node search, in which one begins with a single literature (e.g., that dealing with a disease) and searches for a second unknown literature having complementary knowledge (e.g. that dealing with potential therapies). However, we soon realized that the two-node search is better aligned to the information practices of most biomedical investigators: In this case, the user chooses two literatures A and C and then seeks to identify terms B that occur in the titles of both literatures, that point to meaningful links between them. Thus, when in 2001 we initiated a so-called Phase I grant to demonstrate feasibility of the Arrowsmith search tool, we focused almost exclusively on the two-node search strategy. Marc Weeber, an enthusiastic, visionary informatics researcher [13, 14], gave us crucial assistance at the start of the project; and Vetle Torvik, a brilliant and creative young mathematician [15, 16], has joined us as Project Manager.

2 Project Aims

The Specific Aims of the project are as follows:

1. To test whether Arrowsmith analyses are feasible and useful for assessing research issues, in field tests of neuroscientists working as part of large multi-disciplinary groups; and to incorporate feedback from these users to improve the implementation of the Arrowsmith software.
2. To test whether incorporating MEDLINE record fields other than titles in Arrowsmith analyses will enhance its ability to analyze biomedical literatures.
3. To test whether the free Arrowsmith web site, once upgraded and redesigned with new instructional material, can be made a feasible and useful public forum for conducting Arrowsmith analyses.
4. To test whether Arrowsmith analyses can facilitate inter-laboratory and cross disciplinary collaboration, by identifying complementary sets of investigators that may benefit from working together.

It is most appropriate to discuss our progress on Aims 1-3 together, since our experience with field testers was a major factor in developing both the Arrowsmith website at UIC and the underlying back-end databases. Don has continued to maintain and improve the original website (http://kiwi.uchicago.edu), which has both two-node and one-node search capabilities. However, this site requires users to learn how to upload and download files from the biomedical literature in a particular format, and is limited to 10,000 articles in each set. To overcome these limitations, we created a separate non-mirror site for two-node searches (http://arrowsmith.psych.uic.edu) which is fully interoperable with the popular PubMed interface operated by the National Center for Biotechnology Information (NCBI) – users simply carry out two separate literature searches using the familiar PubMed interface, then click a button and receive a B-list on the webpage within a minute or two. To accomplish this, a dedicated server was set up to handle multi-user queries at UIC; a local copy of MEDLINE was imported, parsed and regularly updated; and the underlying Arrowsmith software was written with fully documented, optimized Perl code. We also programmed a simple, intuitive user interface that field testers found easy to navigate without the need for a tutorial.

The current implementation of the two-node search requires the user to conduct two separate PubMed queries A and C, which define two corresponding sets of articles A and C. (Often the best strategy is to search for articles in which the query term appears in the title. However, the system makes no restriction on how the PubMed queries are conducted; they may involve use of Medical Subject Headings or affiliation fields, may be restricted to review articles, etc.) The Arrowsmith software then stems the titles of the papers in each literature, and makes a list of all single words and two- and three-word phrases that are found in common in the titles of both literatures. Terms that are on a stoplist (consisting of the most common and nonspecific words) are removed, and terms that appear only once in a literature are removed automatically if the literature is larger than 1000 records in size. The resulting "raw B-list" is then filtered and ranked further before being displayed to the user.

Although the field testers obtained a raw B-list quickly and easily, they found it daunting to analyze because for large literatures, there may be hundreds to thousands of terms on the list even after stoplisting. Therefore, a major research effort has been devoted to developing filtering and ranking schemes, so that users can be shown a single short ranked list of the ~50 most promising terms that match their query needs. As summarized below, we have utilized 7 different filtering methods to create a single ranked list. We have two search tracks on the website: a Basic Search option, in which nearly all filters have preset default settings, and an Advanced option in which the user can set filter settings at will.

Filter 1 involves pre-mapping all of the terms (words and up to 3 word phrases) from MEDLINE titles through the National Library of Medicine MetaMap program [17] to identify those that map to one or more semantic categories as defined by the Unified Medical Language System, UMLS [18]. Then, users can examine only those B-terms that fit into one or more desired categories. Because MetaMap cannot optimally recognize terms out of context and because the UMLS is incomplete for some terms (especially protein and gene names), we have added the Tanabe-Wilbur list of predicted gene and protein names [19] as a back-up to identify this semantic category more accurately.

Filter 2 is a frequency filter; users examine B-terms that occur more than (or less than) a certain number of times in either the A or C literature.

Filter 3 is a recency filter; users can examine B-terms that appeared for the first time more recently than (or only earlier than) a given year in either literature.

Filter 4 incorporates information from MEDLINE Medical Subject Headings (MeSH). For each B-term, the MeSH headings of the AB and BC papers are examined [excluding the 20 most frequent MeSH terms in MEDLINE from consideration]. If they have no MeSH terms in common (or fewer than a threshold number), the B-term is removed from the list (Swanson et al., MS submitted for publication).

Filter 5 merges highly related terms within the same semantic category into a single composite B-term, using a statistical model of term co-occurrence within papers (title or abstract fields).

Filter 6 employs "characteristic terms" calculated for the A and C literatures, which are terms in title or abstract fields that occur in that literature significantly more often than in MEDLINE as a whole. Terms that are not characteristic in either literature are removed as being unlikely to be especially significant.

Filter 7 involves cohesion of B-terms: We hypothesize that for any two B-terms, all other things being equal, the one that represents the more narrowly focused literature will be more useful to the user. Thus, we have defined a measure of term cohesion (based on the set of articles in MEDLINE that contain the term in title; Swanson et al. MS submitted) and have pre-calculated cohesion scores for all terms found in MEDLINE titles. When displayed to the user, the most cohesive B-terms are ranked highest and the lowest may be discarded from the list entirely.

3 Field Tester Experiences

Field testers contributed to the development of Arrowsmith in a number of ways. They documented their own “spontaneously conceptualized” two-node searches in entries made in an electronic notebook that contained the details of the search including the query, all edits made and the final B-list, which were sent to UIC. The entries gave the underlying rationale for the search, and rated the ability of the two-node search to find useful information. We also followed-up to track how often the information affected the course of their research in terms of new ideas (that enriched the discussion section of papers or grant proposals), new experiments conducted and new discoveries made. Two 2-day orientation/training sessions were given at UIC (slides of the lectures can be viewed at the Arrowsmith site), and we also visited the field test sites to give demonstrations and lectures. It is important to mention that the field testing sites consist of active neuroscience investigators whose ongoing work generates diverse types of data, ranging from electrophysiology to brain imaging to microarrays to microscopic tissue sections. Moreover, each site is also engaged in their own neuroinformatics research projects. Thus, field testers are in an ideal position to suggest ways that Arrowsmith searching can be adapted specifically in different ways to meet the diverse needs of neuroscientists and other investigators.

We found that even experienced, oriented, well-trained biomedical investigators did not pursue either simple PubMed searches or Arrowsmith searches in the manner envisioned and advocated by information scientists. Most users typically were looking for one or a few, recent papers on a specific topic, and looked for these on the first page of retrieved titles [i.e. the top 20 hits ranked chronologically by PubMed]. They had no interest in finding ALL relevant papers comprehensively. Nor did they attempt to craft their queries carefully -- rather, the queries were deliberately or casually underdetermined and they expected to sift through some irrelevant papers as they scanned quickly through the first few retrieved pages. Their definition of success is quite different from that of an information scientist, and neither recall nor precision (not even top-20 precision) are attuned to this strategy. We are still not quite sure whether this “Googling” strategy is one of naivete or shrewdness!

Field testers did, indeed, find that Arrowsmith assisted them in assessing hypotheses and identifying promising experiments to pursue, and we are currently writing up a paper that will document these findings in detail. A number of the hypotheses assessed by Arrowsmith searches have grown into new research collaborations. However, we found that field testers employed Arrowsmith routinely for three other tasks as well: a) Many of the searches posed for Arrowsmith could have been pursued with simple PubMed searches, but were found convenient to conduct within the Arrowsmith web interface – for example, some users entered two separate searches A and C when desiring to search “A AND C.” b) In addition to seeking conceptual links between two literatures, users often wanted simply to construct a list of items studied in both literatures. c) Many searches were conducted by a user familiar with field A who wanted to browse within an unfamiliar literature C, hoping to find a subset of articles in C relevant to A. We plan to tailor the Arrowsmith interface in the future to support these needs (see below).

4 Progress Towards Facilitating Collaboration

In contrast to Aims 1-3, Aim 4 was considered exploratory, and over the last four years a variety of eclectic approaches have been pursued to develop tools and venues that can assist investigators in finding and facilitating potential collaborations.

To disambiguate author names in MEDLINE, we created a quantitative model, **"Author-ity,"** to estimate the probability that any two papers (sharing the same author last name and first initial) are authored by the same individual [20]. This approach, which will be described below in further detail, has grown into a project to cluster all papers in MEDLINE according to author-individuals. Having such information should assist investigators in finding potential collaborators, particularly in conjunction with planned efforts to create author profiles and to map collaborative networks among people publishing in MEDLINE.

Explicit guidelines were also formulated to help two academic investigators negotiate collaborations [21].

We recognized the need for a forum to create connections among biomedical investigators, who need information management and data integration tools; CS/ informatics researchers, who devise these tools; and social scientists, who evaluate these tools in the context of scientific practice. To create such a forum, a new peer reviewed, open access journal is being launched at BioMed Central, called **Journal of Biomedical Discovery and Collaboration**. I serve as Editor-in-Chief, and William Hersh of Oregon Health Science University is Deputy Editor.

Finally, we invited a prominent social informatics researcher, Carole Palmer of Univ. Illinois-Champaign-Urbana, to analyze the broader information needs and practices of neuroscientists in the field testing sites in detail (participation by field testers was voluntary, but most participated enthusiastically). This project was funded separately as a 3-year project by NSF; the analyses were designed, carried out and are being written up by the UIUC team [22] independently of the Arrowsmith Project.

5 Extending Arrowsmith in New Ways and New Databases

5.1 Incorporating Abstract Terms as B-terms in MEDLINE Searches

Perhaps the single most often-asked question we hear regarding Arrowsmith is, "Why are B-terms taken only from titles?" One reason is that titles of MEDLINE articles are usually very informative, and this maintains a much higher signal-to-noise ratio than simply using terms taken willy-nilly from abstracts and full-text. Another reason is that we already obtain hundreds to thousands of title B-terms from a typical two-node search based on titles. However, there are a number of reasons to include terms in the abstract and full-text of papers as well. First, the title conveys only a small portion of the total information contained in a scientific paper [e.g., 23-25]. Second, terms appearing in the title of a paper may play a qualitatively different role than terms appearing in the abstract – for example, when the term "calpain" appears in the title of a paper, the paper is likely to be studying calpain itself (its enzyme activity, its gene expression, its substrates, etc.). In contrast, when papers contain "calpain" only in abstracts, the authors may be using calpain or calpain inhibitors as experimental reagents. Third, we have found that users often want to examine a B-list, not only to find meaningful conceptual links between two different literatures, but also to quickly construct a list of items that are in common to the two literatures – such items might potentially include affiliations, funding sources, methods, etc., that never appear in the titles of the papers. Finally, there are other approaches to analyzing two literatures that do not involve constructing a B-list for the user at all (see browsing display, below), which may make use of terms in abstract and full-text.

Therefore, a major thrust of our current research is to examine how best to incorporate terms from abstract and full-text into the two-node search. The challenge will be to devise appropriate strategies of restriction, filtering and ranking that elevate the most useful B-terms above a very significant potential "noise" level. These are not simply programming tasks, but involve issues both of basic informatics – how to represent and mine the textual information – and of aligning the display and logical flow of the search process to the needs of scientific users. Handling abstract terms is, in many ways, a straightforward extension of the approaches already outlined for title terms. In contrast, the issue of extracting terms from full-text involves mining PubMed Central rather than MEDLINE, is associated with different user queries, has a number of special problems related to heterogeneity of text portions within an article, and will require establishing a modified user interface.

The basic method of identifying B-terms within abstracts is to scan the abstracts for all terms that are expressed in both A and C literatures. However, if this is performed with no further restrictions, the size of the raw B-list will be an order of magnitude larger than that obtained using title terms alone, and the number of AB and BC papers per B-term will also increase several-fold on average. Therefore, we will only consider terms that occur in at least 2 article titles within MEDLINE (terms that do not occur in titles are mostly broken phrases that arise from stemming artifacts). Then, the list of "raw" B-terms will be filtered and ranked as for B-terms obtained from titles. Even so, we expect that many incidentally mentioned terms will survive the current filtering process, so one or more additional restrictions will be tested and implemented: **1**. B-terms may only be chosen if they occur in the final sentence of the abstract, or within the Conclusion section of a structured abstract. The last sentence often summarizes the main finding of the paper, so this should give the maximal signal-to-noise ratio within the abstract. **2**. In addition to choosing terms from the final sentence, B-terms may also be chosen if they co-occur in the same sentence as a term that occurs in the title of the same paper. **3**. B-terms may also be chosen only if they are characteristic terms of either A or C literatures. Terms that are not characteristic in either A or C literatures are unlikely to convey important, specific information across the disciplines. **4**. When one is interested in identifying terms that may indicate previously unreported links, B-terms that are characteristic in both literatures are probably already well studied. Thus, as a user-specified option for certain purposes, B-terms may only be chosen if they are "characteristic terms" in one literature but not both.

The Arrowsmith tool is generic, and to date we have deliberately refrained from constraining the type of terms or the type of search that a user can perform. Because of this, two persons carrying out the same search on A = "calpain" vs. C = "postsynaptic density" could have entirely different goals in mind: One could be looking for a list of calpain substrates that are located in the postsynaptic density, whereas another could be looking for a list of proteases (other than calpain) that cut postsynaptic proteins. However, an important class of search involves looking for statements that "A affects B" or "B interacts with C." Several tools, including MedMiner [26], Chilibot [27] and BioIE [28] employ part-of-speech information and utilize information regarding interaction verbs to identify sentences that discuss interactions between entities, or even specific types of entities such as genes or proteins. Similar NLP techniques may be employed to assist in identifying relevant B-terms within abstracts.

5.2 Incorporating the Use of Full-Text Terms as B-terms

A local copy of PubMed Central (PMC) can be obtained from NCBI, which includes all papers that are publicly accessible (a.k.a. "open access"). PMC full text articles are XML formatted in a standard manner, and can be parsed to create a database capturing each distinct tagged section in the paper (title, abstract, authors, affiliation, introduction, methods, figure legends, tables and table legends, results, discussion, conclusion, acknowledgments, references). Heuristics do need to be developed to recognize sections in cases where sections are not explicitly tagged (e.g., some journals do not label the Introduction as such, some articles lump Results and Discussion together, and so on). Within each tagged section, text can be split into sentences, so that each sentence will comprise a distinct database entry to facilitate searching of term pairs that co-occur within sentences. Each paper can also be cross-referenced to the paper's PubMed descriptors and (if the paper is also indexed in MEDLINE) to the information encoded in MEDLINE fields. PubMed Central contains 372,000 items as of May 2005, with a total of ~800,000 items expected by the end of 2005.

At least initially, users will employ a separate search interface for full-text queries in PMC. Users will specify two different queries that define two literatures A and C, and the user will be asked to specify a particular type of information to be obtained from a menu of choices:

A) Certain types of information can be processed and presented as B-lists from formally structured fields within the papers, without the need for elaborate filtering and ranking procedures. For example, **author names** common to both literatures can be readily identified from the author field; **affiliations** shared in both literatures, terms used in **acknowledgments** (which may include funding sources and thank-you's to colleagues not named as authors), and d) references cited in both literatures (i.e., **co-citations**).

B) Certain types of information are presented in a more variable form but can be recognized by simple look-up: For example, **reagents or assays** described in methods sections, **anatomical regions** or **diagnostic procedures** mentioned in figure legends, or **genes** listed within tables. Users can specify both the section(s) of the paper to be examined, and the semantic category or nature of the B-term desired from a menu. We can identify the vast majority of such terms by simple look-up, using our existing lists of terms that are mapped to UMLS semantic categories.

C) Finally, we plan to tackle the problem of identifying terms within full-text that can supplement the use of title and abstract terms for making conceptual links across the two literatures. We will only consider terms that occur in at least 2 titles in

MEDLINE (see above), and will avoid sections of the paper such as introduction, methods and discussion where many incidental or historical mentions may occur. Thus, B-terms will only be taken from titles, abstracts, figure legends, tables and table legends, and results (though the user will have the option to add or subtract sections from the list). Additional restrictions will also be implemented, similar to those discussed above for abstract terms in MEDLINE, prior to applying filtering and ranking procedures. Depending on how many instances of a single B-term per paper may be found, and how inclusion of full-text terms affects the size of the B-list, it may be necessary to cluster, compress and/or summarize the sentences related to the same B-term for display to the user.

5.3 Current Challenges

There are at least five limitations in the filtering and ranking of B-terms. **First**, filtering B-terms by UMLS semantic categories does not have ideal flexibility – e.g., one can specify the category of receptor, but not restrict it further to NMDA receptors. **Second**, ranking B-terms by coherence values gives undue weight to very rare B-terms, and we are still learning how to correct this properly. **Third**, the problem of word-sense disambiguation has not been addressed yet. **Fourth**, the default stoplist is approximately 8200 words, which was originally chosen manually by Don Swanson with some further editing at UIC. Because it is difficult to be sure that all of these words are predictably non-interesting to all potential users, we plan to construct a smaller and more rationally chosen stoplist, and we are exploring whether words having extremely low coherence can be fruitfully added to that list. Alternative 1400 and 365 word lists of the most common words are already available within the advanced search settings. **Fifth**, there are both advantages and disadvantages to tokenizing the terms prior to processing (stemming, stoplisting, removing uppercase and splitting into sentences). This speeds processing greatly in two-node searches, but does not allow NLP analyses of free text. At present, the local copy of MEDLINE contains titles represented both in original and tokenized versions; however, the abstracts are only saved in tokenized form, and our term database consists only of tokenized terms. Therefore, new databases will need to be created if we decide to employ information gathered from analyses of free text such as part-of-speech tagging or parsing.

5.4 Adding an Alternative Display for Cross-Disciplinary Browsing

A surprisingly common, yet previously unanticipated, reason that field testers employed the Arrowsmith two-node search was to browse in an unfamiliar discipline, looking broadly for articles that might be relevant to one’s home discipline. In this situation, scrutinizing a B-list is more of a distraction than an aid in identifying relevant papers. Assuming that the user is familiar with literature A, and that the user is not familiar with non-overlapping literature C, the goal is to identify a subset of papers within C that is most closely related to A. Previous studies have employed ontologies or customized standard vocabularies to connect literatures [29]. Certainly MeSH terms could also be used for this purpose. However, MeSH terms may not be ideal for connecting literatures that deal with basic science rather than clinical medicine, and particularly may be too limited in the case of very disparate literatures. For example, shared MeSH terms might not be useful for linking "pesticides" with "fast Fourier transforms." Eventually one would like to be able to connect biomedical articles to literatures found in other fields entirely, such as agriculture, psychology, education and engineering. Thus, just as we have chosen to employ shared B-terms to connect disparate literatures for regular two-node searches, so do we hypothesize that the articles most relevant for scrutiny by a browsing user will be BC, the subset of C that shares certain B-terms with A. Three sub-problems need to be solved in order to create the browsing mode:

1. The size of the subset BC must be chosen so that it represents a relatively small proportion of the C literature. Using the entire raw B-list to define BC would result in a set almost as big as C itself. However, using the default filtering and ranking scheme of the regular two-node search (semantic category, frequency, MeSH and cohesiveness filters) is a promising approach: Using the top 50-100 ranked B-terms results in BC subsets that typically contain only ~200-500 papers. The size of the BC subset is closely linked to the number of B-terms but is relatively insensitive to the size of C. Thus, choosing the top 100 B-terms for a large literature (e.g. schizophrenia, which contains over 50,000 papers) results in a BC subset that represents about 1% of the total. Choosing the optimal size and filtering of the B-list for defining BC is an empirical problem (rather than a theoretical one) and will require trial-and-error testing over a variety of specific searches.

2. A method for clustering the BC papers by topic, and giving a short label to each cluster, needs to be chosen and implemented. Although numerous methods have been explored for thematic clustering, the requirements of clustering in the present context create a number of specific constraints: a) The method needs to be computationally quick, so that it can be computed for thousands of articles within a few seconds. b) Clusters should ideally be "soft"; that is, if individual papers fit several clusters equally well, they can be placed in both. c) The clusters should fit well with the user's conception of how the literature is coarsely organized according to topics. d) Last, but not least, the clusters should be viewable by the user on one webpage. Once the user chooses one cluster, it can be displayed and then optionally re-clustered into another set of subclusters, thus permitting drilling-down of the literature in hierarchical fashion.

The "Anne O'Tate" utility (a separate feature of the Arrowsmith website designed to allow simple data-mining of literatures) currently makes use of MeSH headings in a set-covering approach to form clusters within a set of articles retrieved from PubMed in the following manner: First, all MeSH headings mentioned in the article collection are listed in descending order of frequency. The MeSH that occur in >1/3 of papers are deemed less useful for grouping subclusters, so they are bypassed; for the most frequent MeSH term (below 33%), all papers indexed by that term [and any MeSH terms below that term in the MeSH hierarchy] are placed in cluster #1 and removed from the stack. The MeSH term frequencies are re-calculated for the remaining papers, and the process is repeated to form cluster #2, and so on, until either the clusters contain only single papers or 15 clusters have formed. Any remaining papers [and any papers not indexed with MeSH headings at all] are placed in a final cluster called "other." Finally, for each cluster, a new query is performed containing the original query AND the specific MeSH term defining that cluster – this retrieves the additional papers indexed by that MeSH which had been placed earlier into other clusters, so that individual papers are placed into multiple clusters where appropriate. This method is fast, robust, soft and intuitive, and immediately gives an annotation for the cluster (namely, the MeSH term used to define it).

5.5 Revisiting the One-Node Search

In the Arrowsmith one-node search [9], we begin with a single starting literature A (e.g., the literature on migraine), and compile a list of all terms B_i in the titles of this literature. The list of B_i terms is filtered using a stoplist, and in some cases is further filtered to keep only terms that occur in literature A significantly more than in MEDLINE as a whole. For each B_i term, we search MEDLINE for all papers having B_i in the title [as a practical restriction on the search space, these searches are often restricted to articles sharing a certain MeSH term, such as "Pharmacologic Actions"]. The set of all articles found by searching term B_i is called literature C_i, and all terms in the titles of these papers are referred to as C_i terms. The C_i terms are filtered to keep only terms that occur in literature B_i significantly more than in MEDLINE as a whole, and in some cases, C_i terms are removed if they occur in literature A at all. Then, the C_i terms are combined across all B_i searches to form a master list of C terms. These are ranked according to the number of distinct B terms with which they co-occur -- the presumption is that high ranking C terms are likely to point to previously undocumented, yet biologically meaningful relationships with the A literature.

Many information scientists have explored refinements to the original one node search strategy: Gordon and colleagues used lexical statistics [30] and Latent Semantic Indexing [31] to identify other literatures that contain complementary information. Weeber used UMLS concepts (rather than text words) captured in full-text of articles [13]. Hristovski and Srinivasan have used MeSH terms rather than text words [32, 33], and Wren used manually-constructed "objects" and used a mutual information measure [34-36] to rank objects according to the strength of linkage across literatures A and C. Others, including Pratt [37] and Hearst [38], have explored ways to enhance the user interface to support one-node searching. The Arrowsmith Project offers a public one-node search interface at http://kiwi.uchicago.edu, and Hristovski (http://www.mf.uni-lj.si/bitola/) and Pratt http://litlinker.ischool.washington.edu/) maintain search websites as well, which indicates a high level of interest in these services.

Nonetheless, we deliberately did not study one-node searches as part of the field tester experiences. One concern was that the typical biomedical scientist might not give sufficient credibility to the findings of a one-node search – the indirect links found in the structure of the biomedical literature do not necessarily correspond to the structure of nature itself! Another concern is that the one-node search is an exercise in "searching for an hypothesis," whereas most scientists already have more hypotheses than they can handle, and instead want a tool [the two-node search] to help them assess the ones they already have. Finally, although one-node searches have led to significant testable biomedical predictions, none of the proposed means of filtering and ranking C terms have undergone theoretical or empirical validation. Yet the one-node search can be viewed in several respects as a variant or refinement of the two-node search, and we believe that the time is ripe for tackling the one-node search again.

For example, consider the typical two-node search, in which the user specifies two topical PubMed queries that define literature A and C. One could, instead, input a topical query for literature A [say, "microRNA"], and allow literature C to be all of MEDLINE, or to correspond to a broad MeSH category such as "Disease". This would create an asymmetric situation similar to that envisioned in a one-node search. The current Arrowsmith interface can support literature sizes up to 100,000 in each query, but as an advanced option the user will be allowed to input files of any size, including all of MEDLINE (actually, handling MEDLINE as literature C is an easy task since we have pre-computed frequencies and PubMed IDs for all terms occurring in MEDLINE titles). Conversely, given any two-node search, we could readily construct a ranked list of C terms that show strong, indirect links with literature A. This will provide a different way of browsing an unfamiliar C literature for items that may be relevant to A, which should complement the article-based browsing approach discussed in the previous section. Thus, both the asymmetrical nature of the search, and the construction of a C-list, can be viewed individually as simple extensions of the existing two-node search.

The issue of how to filter and rank C-terms optimally is not easy, in part because different types of searches may have different optimal strategies. For example, in the case where A_i and C_i refer to specific gene names found in the A and C literatures, it is probably true that if A_i and B_i co-occur often in MEDLINE, and B_i and C_I also co-occur more often than expected by chance, then one would expect A_I and C_i to co-occur as well – and if they do not, this raises the question whether this represents an undocumented discovery of a relationship between A_i and C_i. Thus, for predicting gene interactions, co-occurrence frequencies are probably valid for deriving links. However, in other cases, it is probably <u>not</u> valid to focus only on B-terms that occur more often than expected by chance in both literatures: suppose literature A represents the field of microRNAs, and one seeks complementary information in a disparate field (e.g., nutrition). Rather, the terms most likely to point to undocumented discoveries may be those that are characteristic in one, but <u>not</u> both, literatures.

5.6 Gene-Centric Tools

The basic concept of the two-node search can be extended to other datasets such as those in the GEO gene expression database (maintained, like PubMed and PMC, by NCBI as part of their Entrez suite of databases). Suppose that an investigator hypothesizes that two genes A and C should be co-expressed, but they have not been studied together in previous experiments -- one of the genes may have been discovered recently or is an expressed sequence tag (EST), or the two genes may have been studied in different contexts or species and/or were not included on the same microarrays. A two-node search would allow the investigator to find all genes B that were co-regulated with A in certain experiments and that were, separately, co-regulated with C in other experiments. This would allow one to assess whether a relationship between A and C is likely and warrants further study. Alternatively, suppose an investigator has just made a new lab finding that two genes A and C do indeed co-express in one situation. It would be valuable to examine the set of other genes (B genes) that have been reported to co-express with both A and C in different experiments. This B-list will assist in placing the A-C relationship into the larger context of gene networks. One may also wish simply to combine data across multiple experiments of the same type. For example, at present, 19 different experiments in the GEO database have examined the expression of eIF2c2 in human brain. (These experiments are not necessarily all comparable to each other, but one could filter them manually if desired.) Making a list of genes that co-express with eIF2c2 across multiple experiments is one way of detecting the genes that are most robustly linked with eIF2c2. Conversely, one might want to compare two apparently disparate experiments and find gene pairs that are co-regulated similarly in both cases.

The GEO Gene Expression database can be adapted for conducting two-node searches in a manner that is analogous to the text-based Arrowsmith search: The user search-interface would replicate the NCBI site (GEO Profiles). A two-node search might go something like this: **1**. The user inputs the name of a gene, together with additional restrictions such as platform, species, tissue or developmental stage. This request is processed to retrieve all GEO Profiles that satisfy the query, giving **literature A**. (A GEO Profile describes a single experiment involving that gene.) Automatically, for each GEO Profile having more than 2 experimental conditions, the software computes all of the "profile neighbors" of that gene – this computation uses the Pearson correlation coefficient to identify a set of the other genes whose expression was most similar to the index gene in that same experiment. Profile neighbors are calculated using a Pearson linear correlation with a threshold of 0.7, and a t-test with an arbitrarily-determined Bonferroni adjustment - the top 100 profile neighbors are presented. **2**. The user inputs the name of a second gene, together with restrictions as desired, which gives **literature C**. The GEO Profiles are retrieved for the second gene, and the "profile neighbors" are computed. **3**. All "profile neighbors" which are common to both literatures are identified and displayed on a single **B-list** of gene names. **4**. The user can select any B-term and see the Profiles that include A and B, juxtaposed to the Profiles that include B and C.

This scheme is quite analogous to the situation of searching two literatures in PubMed. At present, the GEO database is small and extremely heterogeneous, so that it is not easy to formulate useful A and C searches. This limitation should become less important with time, as GEO becomes more populated and as the scientific community formulates standard platforms and standard formats for documenting experiments.

6 Spin-Offs Supported by the Arrowsmith Project

6.1 "Anne O'Tate"

We have programmed a utility on the main Arrowsmith homepage that displays, for any collection of PubMed articles, a ranked list of most frequent terms, most frequent MeSH headings, and most "important" words appearing in title or abstract. The utility also displays most frequent author names, affiliations, journals, a histogram of years of publication, and a list of the terms that have appeared for the first time most recently in MEDLINE. Also, the user can cluster the articles into topical subgroups (as discussed above). Each of these lists gives a different, partially complementary summary view of the contents of the article collection.

6.2 "WETLAB"

A simple open source electronic laboratory notebook has been programmed in Java, that is oriented to the needs of wet-lab neuroscientists. This notebook, **WETLAB**, allows flexible searching of both data and metadata across templated text fields, stores the text in XML files, and allows data-sharing by ftp or email. WETLAB is currently undergoing beta testing and will be placed on the Arrowsmith website for unrestricted download.

6. 3 Genomics Studies

Unlike the other Arrowsmith projects, this work has not been directed (yet) towards generating a software tool or web service. Rather, we have utilized some of our data-mining approaches to analyze the newly discovered class of genes known as microRNAs and their targets in the mammalian genome [39-43]. This combined computational and wet-lab project involves several of the Arrowsmith field testers and is an important scientific test bed for tool development, as well as an exciting scientific arena in its own right.

6.4 "Author-ity"

As a first step towards creating an author-individual database of all articles in MEDLINE, we have created a statistical model of how two papers authored by the same vs. different individuals vary on a similarity profile computed across different MEDLINE attributes (title words, MeSH, co-author names, affiliation words, etc) [20]. We have programmed a tool, "Author-ity," that resides on the main Arrowsmith homepage: the user specifies a (last name, first name, optional middle initial and optional suffix), and retrieves a list of papers bearing that name. The user then chooses one paper from the list and obtains a ranked list of all of the other papers in descending order of probability that the paper was written by the same individual.

A monotone model is satisfied when the value of the function increases monotonically as the value for a given variable increases (and all other variables' values remain the same). Such functions are easily computed and can place multi-dimensional data onto a single dimensional ranking score or probability value in a manner that takes into account nonlinear and interactive effects across dimensions, yet is readily interpretable for the nature and contribution of each dimension. This type of model appeared to be ideal for the task of comparing two different articles in MEDLINE bearing the same author name, and asking whether they were authored by the same individual.

First, we hypothesized that different papers written by the same individual will tend to share certain characteristic features, not only dealing with the author's personal information (name and affiliation attributes) but other attributes of the articles as well. The probabilistic model [20] describes, for any two papers bearing the same author (last name, first initial), how similar the two papers are across 8 different dimensions: middle initial match, suffix match (e.g., Jr. or III), journal name, language of article match, number of co-author names in common, number of title words in common after preprocessing and removing *title-stopwords*, number of affiliation words in common after preprocessing and removing *affiliation-stopwords*, and number of MeSH words in common after preprocessing and removing *mesh-stopwords*. These are calculated solely from comparing corresponding MEDLINE fields. The resulting 8-dimensional comparison vector, which we call the "similarity profile," is computed for the members of two large reference sets – a match set, consisting of many (millions) pairs of papers very likely to be co-authored by the same individual across MEDLINE, and a non-match set consisting of many pairs of papers known to be authored by different individuals. These training sets were very robust against inclusion of incorrect data.

Thus, given any pair of papers bearing the same author (last name, first initial), we compute the similarity profile and observe its relative frequency in the match set vs. the non-match set. If the observed profile is much more frequent in the match set than in the non-match set, it is likely that the two papers were written by the same individual. The ratio of the profile frequency in the match vs. non-match sets, together with an estimate of the *a priori* probability that any two randomly chosen papers having that name will be authored by the same individual, gives an estimate of the probability that the two papers were written by the same individual [20]. We plan to employ clustering algorithms on papers bearing the same (last name, first initial) to form clusters of papers that can be assigned to distinct author-individuals across MEDLINE.

If monotone models are so wonderful, why aren't they utilized more often in a variety of other situations in medical informatics and bioinformatics, for example, to improve algorithms for information retrieval? Possibly the reason is that it is often hard to generate enough training data to properly fit a monotone model, especially when the number of distinct observable cases is high (e.g., when there are many variables or variables are continuous). Hopefully the use of massive, automatically generated training sets should enhance the popularity of this approach.

7 Conclusions

Don, Vetle and I differ markedly in our backgrounds and personalities, yet are compatible in terms of our general approach to informatics, and this has given a distinct flavor to our joint research efforts:

First, are interested in having computers do what they do best, rather than what people do best. We are not against AI, NLP or machine learning approaches. However, our own goal is to create tools that extend (but not replace) the normal capabilities of people. We seek to make telescopes, not artificial retinas.

Second, we have undertaken a commitment to developing free, public tools. The Arrowsmith websites require no passwords or registration, and although they are under continual development, they are not simply demonstration sites but offer full-strength capabilities for the real-life information needs of scientists.

Third, the tools that we develop are very simple and generic. They are applicable to all fields of biomedical science, by scientists at all levels of seniority, and equally by people running small laboratories or practitioners of Big Science.

Fourth, the field testers are not simply beta testers, experts or “users” but are true scientific collaborators in the development process. It is common in bioinformatics to combine computational biology and wet-lab studies, but I think that the Arrowsmith project has a uniquely multi-disciplinary discovery process that encourages investigators to contemplate radically new directions in their research.

Fifth, we are attuned to a paradoxical requirement of informatics tools: they need to be designed to align well with the perceived needs of scientists and their daily practice, yet the tools also need to be designed to expand scientists’ horizons – to improve their ability to handle information and scientific ideas, and to raise expectations and consciousness in a manner that will reshape routine scientific practice [44].

The Arrowsmith Project has demonstrated that it is feasible for scientific investigators to conduct two-node searches in their daily lives. The next challenge is to publicize the tool widely and to induce young scientists, especially, to think explicitly about how they formulate and assess new hypotheses.

Acknowledgments

Supported by NIH grants LM07292 and LM08364. This Human Brain Project Neuroinformatics research is funded jointly by the National Library of Medicine and the National Institute of Mental Health. I thank Marc Weeber, Alan Lian, Wei Zhou, Wei Zhang and Clement Yu for contributions in computer science and informatics, and Amanda Bischoff-Grethe, Lauren Burhans, Christopher Dant, Mike Gabriel, Ramin Homayouni, Alireza Kashef, Maryann Martone, Lauren Penniman, Guy Perkins, Diana Price, Allan Reiss and Andrew Talk for their participation in this collaborative venture as field testers.

Reference

1. Swanson DR. Fish oil, Raynaud's Syndrome, and undiscovered public knowledge. Perspect. Biol. Med. 1986; 30: 7-18.
2. Swanson DR. Undiscovered public knowledge. Library Q 1986; 56: 103-118.
3. Swanson DR. Two medical literatures that are logically but not bibliographically connected. JASIS 1987; 38: 228-233.
4. Swanson DR. Migraine and magnesium: eleven neglected connections. Perspect. Biol. Med. 1988; 31: 526-557.
5. Smalheiser NR, Swanson DR. Assessing a gap in the biomedical literature: magnesium deficiency & neurologic disease. Neurosci. Res. Commun. 1994; 15: 1-9.
6. Smalheiser NR, Swanson DR. Linking estrogen to Alzheimer's Disease: an informatics approach. Neurology 1996; 47: 809-810.
7. Smalheiser NR, Swanson DR. Indomethacin and Alzheimer s Disease. Neurology 1996; 46: 583.

8. Smalheiser NR, Swanson DR. Calcium-independent phospholipase A2 and schizophrenia. Arch. Gen. Psychiat. 1998; 55: 752-753.
9. Swanson DR, Smalheiser NR. An interactive system for finding complementary literatures: a stimulus to scientific discovery. Artif. Intelligence 1997; 91: 183-203.
10. Smalheiser NR, Swanson DR. Using ARROWSMITH: a computer-assisted approach to formulating and assessing scientific hypotheses. Computer Methods and Programs in Biomedicine 1998; 57: 149-153.
11. Smalheiser NR. Predicting emerging technologies with the aid of text-based data mining: a micro approach. Technovation 2001; 21: 689-693.
12. Swanson DR, Smalheiser NR, Bookstein A. Information discovery from complementary literatures: categorizing viruses as potential weapons. JASIST 2001; 52: 797-812.
13. Weeber M, Vos R, Baayen RH. Using concepts in literature-based discovery: Simulating Swanson's raynaud - fish oil and migraine - magnesium discoveries. JASIST 2001; 52: 548-557.
14. Weeber M, Vos R, Klein H, De Jong-Van Den Berg LT, Aronson AR, Molema G. Generating hypotheses by discovering implicit associations in the literature: a case report of a search for new potential therapeutic uses for thalidomide..JAMIA 2003; 10: 252-9.
15. Torvik VI, Triantaphyllou E. Guided Inference of Nested Monotone Boolean Functions. Information Sciences 2003; 151: 171-200.
16. Torvik VI, Triantaphyllou E. Discovering rules that govern monotone phenomena. In Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques (Triantaphyllou and Felici, eds.) Massive Computing Series, Springer, 2005, Chapter 4: 149-192, in press.
17. Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc AMIA Symp. 2001;:17-21.
18. Lindberg DA, Humphreys BL, McCray AT. The Unified Medical Language System. Methods Inf Med. 1993 Aug;32(4):281-91. Related Articles, Links
19. Tanabe L, Wilbur WJ. Generation of a large gene/protein lexicon by morphological pattern analysis. J Bioinform Comput Biol. 2004 Jan;1(4):611-26.
20. Torvik VI, Weeber M, Swanson DR, Smalheiser NR. A probabilistic similarity metric for MEDLINE records: a model for author name disambiguation. JASIST 2005; 56(2): 140-158.
21. Smalheiser NR, Perkins GA, Jones S. Guidelines for negotiating scientific collaborations. PLoS Biology 2005; 3(6): e217.
22. Palmer CL, Cragin MH, Hogan TP. Information at the Intersections of Discovery: Case Studies in Neuroscience. Proc. ASIST annual meeting. 2004 Nov; 448-455.
23. Kostoff RN, Block JA, Stump JA, Pfeil KM. Information content in MEDLINE record fields. Int J Med Inform. 2004 Jun 30;73(6):515-27.
24. Ding J, Berleant D, Nettleton D, Wurtele E. Mining MEDLINE: abstracts, sentences, or phrases? Pac Symp Biocomput. 2002;:326-37.
25. Shah PK, Perez-Iratxeta C, Bork P, Andrade MA. Information extraction from full text scientific articles: where are the keywords? BMC Bioinformatics. 2003;4:20.
26. Tanabe L, Scherf U, Smith LH, Lee JK, Hunter L, Weinstein JN. MedMiner: an Internet text-mining tool for biomedical information, with application to gene expression profiling. Biotechniques. 1999 Dec;27(6):1210-4, 1216-7.
27. Chen H, Sharp BM. Content-rich biological network constructed by mining PubMed abstracts. BMC Bioinformatics. 2004 Oct 8;5(1):147.
28. Divoli A, Attwood TK. BioIE: extracting informative sentences from the biomedical literature. Bioinformatics. 2005 May 1;21(9):2138-9.

29. Chen H, Martinez J, Ng TD, Schatz BR. A concept space approach to addressing the vocabulary problem in scientific information retrieval: An experiment on the worm community system. 1997 Jan. JASIST 48 (1):17-31.
30. Lindsay RK, Gordon MD. Literature-based discovery by lexical statistics. JASIS 1999; 50: 574-587.
31. Gordon MD, Dumais S. Using latent semantic indexing for literature based discovery. JASIS 1998; 49: 674-685.
32. Hristovski D, Peterlin B, Mitchell JA, Humphrey SM. Using literature-based discovery to identify disease candidate genes. Int J Med Inform. 2005 74:289-98.
33. Srinivasan P Text Mining: Generating Hypotheses from MEDLINE JASIST 2004; 55(5): 396-413.
34. Wren JD, Bekeredjian R, Stewart JA, Shohet RV, Garner HR. Knowledge discovery by automated identification and ranking of implicit relationships. Bioinformatics. 2004 Feb 12;20(3):389-98.
35. Wren JD, Garner HR. Shared relationship analysis: ranking set cohesion and commonalities within a literature-derived relationship network. Bioinformatics. 2004 20:191-8.
36. Wren JD. Extending the mutual information measure to rank inferred literature relationships. BMC Bioinformatics. 2004 Oct 7;5(1):145.
37. Pratt W, Yetisgen-Yildiz M. LitLinker: Capturing Connections across the Biomedical Literature. Proceedings of the International Conference on Knowledge Capture (K-Cap'03). p. 105-112. Florida, October 2003.
38. Hearst MA. Untangling text data mining. Proc. Assoc. Comp. Ling. 1999.
39. Smalheiser NR. EST analyses predict the existence of a population of chimeric microRNA precursor-mRNA transcripts expressed in normal human and mouse tissues. Genome Biology 2003; 4:403.
40. Smalheiser NR, Torvik VI. A population-based statistical approach identifies parameters characteristic of human microRNA-mRNA interactions. BMC Bioinformatics 2004;5:139.
41. Smalheiser NR, Torvik VI. Mammalian microRNAs derived from genomic repeats. Trends in Genetics 2005; 21(6): 322-326.
42. Smalheiser NR, Torvik VI. Complications in mammalian microRNA target prediction. To be published in "MicroRNA: Protocols", ed. S.-Y. Ying, in the series "Methods in Molecular Biology", Humana Press, 2005.
43. Lugli G, Larson J, Martone ME, Jones Y, Smalheiser NR. Dicer and eIF2c are enriched at postsynaptic densities in adult mouse brain and are modified by neuronal activity in a calpain-dependent manner. J. Neurochem. 2005, in press.
44. Smalheiser NR. Informatics and hypothesis-driven research. EMBO Reports 2002; 3: 702.

Practical Algorithms for Pattern Based Linear Regression

Hideo Bannai[1], Kohei Hatano[1], Shunsuke Inenaga[1,2], and Masayuki Takeda[1,3]

[1] Department of Informatics, Kyushu University,
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan
{bannai, hatano, shunsuke.inenaga, takeda}@i.kyushu-u.ac.jp
[2] Japan Society for the Promotion of Science
[3] SORST, Japan Science and Technology Agency (JST)

Abstract. We consider the problem of discovering the optimal pattern from a set of strings and associated numeric attribute values. The goodness of a pattern is measured by the correlation between the number of occurrences of the pattern in each string, and the numeric attribute value assigned to the string. We present two algorithms based on suffix trees, that can find the optimal substring pattern in $O(Nn)$ and $O(N^2)$ time, respectively, where n is the number of strings and N is their total length. We further present a general branch and bound strategy that can be used when considering more complex pattern classes. We also show that combining the $O(N^2)$ algorithm and the branch and bound heuristic increases the efficiency of the algorithm considerably.

1 Introduction

Fundamental biological molecules such as DNA, RNA, and proteins can be regarded as strings over a certain alphabet. Although the whole genomic sequences of many species are now becoming available, there is still much that is unknown about the information that lie hidden in them. Computational analysis of these sequences rely on the principle that similarity as strings implies similarity in their sequence structure, which in turn implies similarity in their functions. Therefore, methods for efficiently and effectively discovering meaningful *patterns*, or sequence elements which are conserved in a given set of strings, is an important problem in the field of Bioinformatics [1].

Earlier work on pattern discovery focus on discovering the most conserved pattern in a given set of strings, generally preferring *longer* patterns which occur in *most* of the sequences in the set. Another situation is when we are given two sets of strings, where one set (positive set) consists of sequences known to possess some biological characteristic, while the other (negative set) consists of sequences known not to posses these characteristics. The problem is to find a discriminating pattern, that is, a pattern which occurs in most strings of the positive set, but does *not* occur in most of the strings of the negative set [2, 3, 4, 5, 6].

Recently, there have been several works which incorporate numeric attributes which are obtained from other sources, e.g. gene expression data obtained from

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 44–56, 2005.

microarray experiments, in order to find meaningful patterns more effectively [7, 8, 9, 10, 11]. The basic idea of these methods is to find sequences elements whose occurrences in the sequences are correlated with the numeric attributes. For example, gene expression is regulated by molecules called *transcription factors*, which bind to specific sequences usually in the upstream of the coding region of a gene. The binding sites for a given transcription factor are fairly conserved across genes which are regulated by the same transcription factor. Therefore, if we can find sequence elements which occur in upstream regions of genes which are relatively highly expressed, while not occurring in upstream regions of genes whose expression is relatively low (or vice versa), such patterns are likely to be binding sites of specific transcription factors.

In [7], substring patterns of up to length 7 are scored according to the linear fit between the number of occurrences in the upstream region and the expression level of the gene. However, they do not consider any algorithm for solving the problem efficiently. Also, the choice of the maximum pattern length is arbitrary and it is not guaranteed that the optimal pattern will be found. Algorithmic work for solving a similar problem has been considered in [8, 9]. In this problem setting, the number of occurrences of a pattern is only considered as an indicator value of 0 or 1, i.e. whether the pattern occurs in the string or not. Based on the algorithm for solving the color set size problem [12], a very efficient $O(N)$ time algorithm for finding the optimal substring pattern in this problem setting is given in [9]. A general branch and bound strategy that can be used for more complex patterns where the problem can be NP-hard, is given in [8].

Although the algorithms above have been shown to discover similar motifs as in [7], it is generally believed that multiple occurrences of a binding site motif in the upstream region of a gene will strengthen the function of the transcription factor for that gene. In this paper, we give an efficient algorithm to discover the optimal pattern, taking into account the number of occurrences of the pattern in each string, as in the problem setting in [7]. We first present two simple algorithms based on the suffix tree data structure that finds the optimal substring pattern (without a restriction in the length of the pattern), respectively in $O(nN)$ and $O(N^2)$ time. We further develop and apply a branch and bound strategy in order to speed up the algorithm, also allowing the problem to be solved for more complex and descriptive classes of patterns. The algorithms developed are applied to real biological data to show the efficiency and effectiveness of the approach.

2 Preliminaries

Let Σ be a finite alphabet. An element of Σ^* is called a *string*. Strings x, y, and z are said to be a *prefix*, *substring*, and *suffix* of string $w = xyz$, respectively. The length of a string w is denoted by $len(w)$. The empty string is denoted by ε, that is, $len(\varepsilon) = 0$. For any set S, let $|S|$ denote the cardinality of the set. The empty set is denoted by $\emptyset$, that is, $|\emptyset| = 0$. Let $\mathbf{R}$ represent the set of real numbers.

Let Π be a set of *patterns.* We call a function defined over a text string and pattern $\psi : \Sigma^* \times \Pi \to \mathbf{R}$ a *matching function.* Let $\psi_p : \Sigma^* \to \mathbf{R}$ represent the matching function for a fixed $p \in \Pi$, that is, $\psi(s,p) = \psi_p(s)$ for any text string $s \in \Sigma^*$. For the matching function value, we shall consider the number of occurrences of a given pattern in the text string. A *substring pattern* p is a pattern $p \in \Pi = \Sigma^*$, where the matching function value $\psi_p(s)$ is defined as the number of substrings in s which is equal to p. A *don't care pattern* p is a pattern $p \in \Pi = (\{.\} \cup \Sigma)^*$, where "." is a don't care symbol, and the matching function value $\psi_p(s)$ is defined as the number of substrings in s which can be obtained from p by appropriate substitution of the don't care symbols with characters of the alphabet Σ. For the above two pattern classes, we shall refer to Σ or $\{.\} \cup \Sigma$ as the *pattern alphabet.*

For the rest of the paper, we assume that we are given as input, a sequence of ordered pairs consisting of a string and an associated numeric attribute value: $\{(s_1, y_1), \ldots, (s_n, y_n)\} \subset \Sigma^* \times \mathbf{R}$. Let $N = \sum_{i=1}^{n} len(s_i)$ represent the total length of the input strings. We denote by $\boldsymbol{y} = (y_1, \ldots, y_n)^T \in \mathbf{R}^n$, the vector consisting of the numeric attribute values. Further, for a given pattern p, we denote by $\boldsymbol{\psi_p(s)} = (\psi_p(s_1), \ldots, \psi_p(s_n))^T \in \mathbf{R}^n$, the vector consisting of the matching function values for the input text strings. We define for later use, a preorder over patterns as follows:

Definition 1. *For any $p', p \in \Pi$, denote $p' \preceq p$ if for all $\{s_i \mid i = 1, \ldots, n\}$, $\psi(s_i, p') \leq \psi(s_i, p)$.*

We now consider how to score the *goodness* of a given pattern. For a given $\boldsymbol{x} = (x_1, \ldots, x_n)^T \in \mathbf{R}^n$, let $\boldsymbol{X} = (\mathbf{1}, \boldsymbol{x})$, and define

$$RSS(\boldsymbol{y}|\boldsymbol{x}) = ||\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}||^2 = \sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$

where $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)^T$ are the least square estimates of $\boldsymbol{\beta}$ in the linear model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

We consider the problem of finding the pattern which can best fit the numeric attribute values y_i with respect to $\psi_p(s_i)$.

Definition 2 (Pattern Based Linear Regression). *We define the pattern based linear regression problem as follows. Given $\{(s_1, y_1), \ldots, (s_n, y_n)\} \subset \Sigma^* \times \mathbf{R}$, and a matching function ψ, find the pattern $p \in \Pi$ that minimizes*

$$RSS(\boldsymbol{y}|\boldsymbol{\psi_p(s)}) = ||\boldsymbol{y} - (\mathbf{1}, \boldsymbol{\psi_p(s)})\hat{\boldsymbol{\beta}}||^2 = \sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 \psi_p(s_i)))^2$$

where $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)^T$ are the least square estimates of $\boldsymbol{\beta}$ in the linear model

$$\boldsymbol{y} = (\mathbf{1}, \boldsymbol{\psi_p(s)})\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

We note that the consistency problem [13,14] is a special case of our problem. Since the consistency problem is shown to be NP-complete for several pattern classes (e.g. *subsequence patterns*), the above problem is NP-hard for such cases. An exception is the case for the substring pattern class, for which we shall present efficient solutions in Section 3.

3 Methods

3.1 Finding the Optimal Substring Pattern

When considering substring patterns, it can be shown that the number of possible patterns which give distinct *RSS* scores is linear in the total length of strings, i.e. $O(N)$. We make use of a very convenient and well studied data structure called *suffix trees*.

Generalized Suffix Trees. A *suffix tree* [15] for a given string s is a rooted tree whose edges are labeled with substrings of s, satisfying the following characteristics. For any node v in the suffix tree, let $l(v)$ denote the string spelled out by concatenating the edge labels on the path from the root to v. For each leaf node v, $l(v)$ is a distinct suffix of s, and for each suffix of s, there exists such a leaf v. Furthermore, each node has at least two children, and the first character of the labels on the edges to its children are distinct. A generalized suffix tree (GST) for a set of n strings $S = \{s_1, \ldots, s_n\}$ is basically a suffix tree for the string $s_1\$_1 \cdots s_n\$_n$, where each $\$_i$ $(1 \leq i \leq n)$ is a distinct character which does not appear in any of the strings in the set. However, all paths are ended at the first appearance of any $\$_i$, and each leaf is labeled with id_i. An example of a GST is shown in Fig. 1. It is well known that suffix trees (and generalized suffix trees) can be represented in linear space and constructed in linear time [15] with respect to the length of the string (total length of the strings for GST).

Notice that candidate substring patterns may be restricted to those represented by nodes of the generalized suffix tree. This is because, for any substring pattern that does not correspond to a path in the suffix tree, the pattern does not occur in any of the strings in the set. Also, note that for a given pattern that does correspond to a path in the suffix tree, all occurrences of the pattern in the strings are represented by the leaves of the suffix tree in the subtree below this path. This means that for any substring pattern that corresponds to a path that ends in the middle of an edge of the suffix tree, its occurrences in the strings are identical to the occurrences of the substring pattern corresponding to the path extended to the next node.

As stated in the Introduction, the pattern based linear regression problem has been shown to be solvable in $O(N)$ time if the matching function is considered to be an indicator function returning the value 0 or 1 [9]. However, it is assumed that the score of a pattern is a function of the sum of the matching function values and the sum of the numeric attribute values of the strings that the pattern occurs in. The algorithm cannot be applied to our case since we

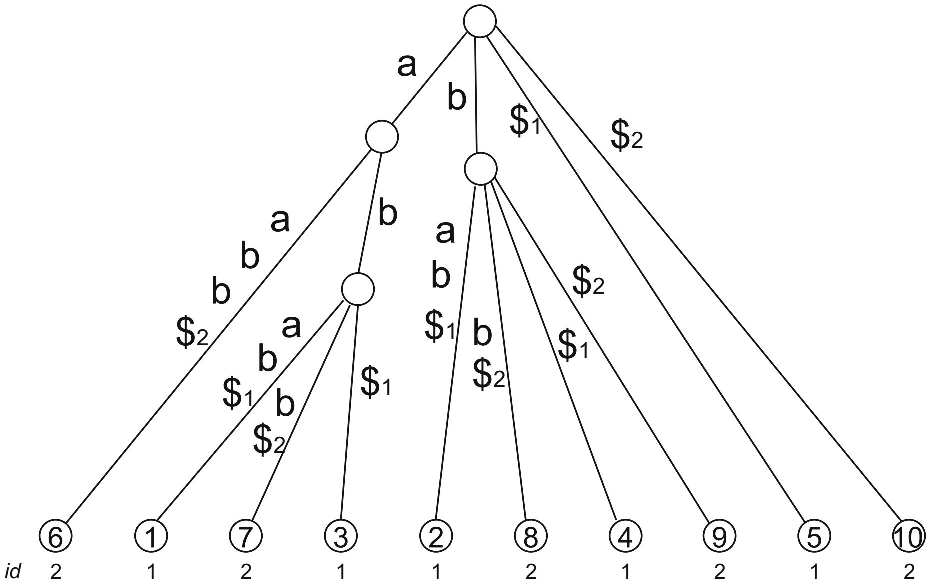


Fig. 1. Example of a generalized suffix tree for the string set { `abab`, `aabb` }

require the matching function values of each string in order to compute the RSS score. Below, we give two algorithms which calculate the score for each candidate pattern p corresponding to a node in the generalized suffix tree.

An $\boldsymbol{O(N^2)}$ Algorithm. Calculating the score for each of the $O(N)$ candidate patterns requires the calculation of $\boldsymbol{\psi_p(s)}$ for each p, as well as the calculation of the least square estimates. The former can be calculated in $O(N)$ time using well known linear time string pattern matching algorithms such as the Knuth-Morris-Pratt algorithm [16]. The latter can be calculated by first obtaining the least square estimate of the parameters:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y},$$

where $\boldsymbol{X} = (\mathbf{1}, \boldsymbol{\psi_p(s)})$. It is not difficult to see that this can be calculated in $O(n)$ time. Further, $RSS(\boldsymbol{y}|\boldsymbol{\psi_p(s)})$ can be calculated in $O(n)$ time, and therefore the resulting time complexity is $O(N) \cdot O(N + n) = O(N^2)$.

An $\boldsymbol{O(Nn)}$ Algorithm. Consider assigning a vector of length n at each node and leaf of the suffix tree. The vectors are initialized as follows: for an internal node, all values are set to 0. For a leaf labeled with id_i, the value at the ith position is set to 1, and the rest is set to 0. Then, with a bottom-up (postorder) traversal on the suffix tree, we add up the values in the vector at each node, element-wise, into the vector of its parent node. The result is that we obtain, at each node, a vector of length n where each position i of the vector represents the number of leaves in the subtree of the suffix tree, with id_i. This corresponds to the number of times the substring pattern occurs in string s_i, and consequently, the vector represents $\boldsymbol{\psi_p(s)}$. This means that $\boldsymbol{\psi_p(s)}$ can be calculated in $O(n)$ time for each pattern, resulting in a total of $O(Nn)$ time for the score calculations.

3.2 Branch and Bound Strategy

Since the pattern based linear regression problem can be NP-hard when considering more complex pattern classes, we propose an enumerative branch and bound framework for finding the optimal pattern. The basic idea of the enumeration is similar to previous works $[3, 4, 2, 5, 6, 8]$. The main contrivance of this paper is in the method for calculating the lower bound of the *RSS* score for specific subspaces of the pattern space.

The algorithm proceeds by traversing and enumerating nodes on a *search tree*, where each node in the tree represents some pattern in Π. For any pattern p in the search tree, let p' be a pattern represented by a node in the subtree rooted at the node for p. While traversing the search tree at the node corresponding to p, suppose that we are able to calculate a lower bound for the *RSS* for any pattern p'. If this lower bound is greater than the current best *RSS* found in the traversal, we know that the score for p' cannot be below the current best *RSS*. This allows us to prune the search space by disregarding the subtree of the search tree rooted at the node corresponding to p.

Below, we show how such a lower bound can be calculated. The assumptions for our calculations below is that $p' \preceq p$. For the case of string patterns and don't care patterns, this assumption can be fulfilled by considering the search tree described as follows. The root corresponds to the empty string ε, and each node v will have child nodes whose pattern corresponds to the pattern obtained by extending the pattern at node v by one character of the pattern alphabet. Although we do not elaborate in this paper, the same branch and bound approach can be applied to a variety of other patterns such as approximate patterns and degenerate patterns.

Problem 1 (lower bound of the residual sum of squares for $p' \preceq p$). Given some pattern $p \in \Pi$, find a lower bound of the score function $RSS(\boldsymbol{y}|\boldsymbol{\psi_{p'}(s)})$ for any pattern $p' \preceq p$.

Below, let $\boldsymbol{x} = (x_1, \dots, x_n)^T = \boldsymbol{\psi_p(s)}$. Also, let $\boldsymbol{x}^{\text{opt}} = (x_1^{\text{opt}}, \dots, x_n^{\text{opt}})^T = \arg\min_{\boldsymbol{z} \in D_x} RSS(\boldsymbol{y}|\boldsymbol{z})$ where $D_x = \{(z_1, \dots z_n)^T \mid 0 \le z_i \le x_i,\ i = 1, \dots, n\}$, and let $\hat{\boldsymbol{\beta}}^{\mathbf{opt}} = (\hat{\beta}_0^{\text{opt}}, \hat{\beta}_1^{\text{opt}})$ be the least square estimates for the linear model $\boldsymbol{y} = (\mathbf{1}, \boldsymbol{x}^{\text{opt}})\boldsymbol{\beta} + \boldsymbol{\epsilon}$. Our objective now is to find $RSS(\boldsymbol{y}|\boldsymbol{x}^{\text{opt}})$. This can be considered as a relaxed version of the problem stated above, since we do not require that there exists a pattern $p' \preceq p$ such that $\boldsymbol{\psi_{p'}(s)} = \boldsymbol{x}^{\text{opt}}$. The following theorem gives a simple lower bound on the *RSS* score.

Algorithm 1: Simple algorithm for calculating a lower bound of RSS.

```
Input: $\boldsymbol{x} = (x_1, \ldots, x_n)^T$,$\boldsymbol{y} = (y_1, \ldots, y_n)^T$
Output: $\mathsf{lb} \leq \min\{RSS(\boldsymbol{y}|\boldsymbol{z}) \mid \boldsymbol{z} = (z_1, \ldots, z_n)^T, 0 \leq x'_i \leq x_i\}$
$X \leftarrow \{i \mid x_i = 0\}$;
if $X = \emptyset$ then return 0;        /* Can move all $x_i$ onto a single line. */
$\mathsf{m} \leftarrow \sum_{i \in X} y_i / |X|$;                /* mean of values $\{y_i \mid x_i = 0\}$ */
$\mathsf{lb} = \sum_{i \in X} (\mathsf{m} - x_i)^2$;   /* residual sum of squares for points $\{x_i = 0\}$ */
return lb
```

Theorem 1 (simple lower bound). *Let* $X = \{i \mid x_i = 0\}$. *If* $X \neq \emptyset$, *then*

$$\sum_{i \in X} (m - y_i)^2 \leq RSS(\boldsymbol{y}|\boldsymbol{x}^{\mathrm{opt}})$$

where $m = \sum_{i \in X} y_i / |X|$.

Proof. For any i, if $x_i = 0$, then since $0 \leq x_i^{\mathrm{opt}} \leq x_i$ we have that $x_i^{\mathrm{opt}} = 0$. Ignoring the residuals for all data points $x \neq 0$, the minimum possible residual sum of squares for data where $x_i = x_i^{\mathrm{opt}} = 0$ must be smaller than the residual sum of squares for the entire data set. □

The simple lower bound can be calculated with the pseudo-code shown in Algorithm 1.

Next, we try to improve on this lower bound. The following lemma gives the conditions between $\boldsymbol{x}^{\mathrm{opt}}$ and the regression line.

Lemma 1. *For all* $i = 1, \ldots, n$, *if* $\hat{\beta}_1^{\mathrm{opt}} > 0$ *then*

$$x_i^{\mathrm{opt}} = \begin{cases} 0 & \text{if } x_i = 0 \text{ or } y_i \leq \hat{\beta}_0^{\mathrm{opt}} \\ (y_i - \hat{\beta}_0^{\mathrm{opt}}) / \hat{\beta}_1^{\mathrm{opt}} & \text{otherwise } (x_i > 0 \text{ and } y_i > \hat{\beta}_0^{\mathrm{opt}}) \end{cases}$$

and if $\hat{\beta}_1^{\mathrm{opt}} < 0$,

$$x_i^{\mathrm{opt}} = \begin{cases} 0 & \text{if } x_i = 0 \text{ or } y_i \geq \hat{\beta}_0^{\mathrm{opt}} \\ (y_i - \hat{\beta}_0^{\mathrm{opt}}) / \hat{\beta}_1^{\mathrm{opt}} & \text{otherwise } (x_i > 0 \text{ and } y_i < \hat{\beta}_0^{\mathrm{opt}}). \end{cases}$$

Proof. We will prove the case for $\hat{\beta}_1^{\mathrm{opt}} > 0$. The case for $\hat{\beta}_1^{\mathrm{opt}} < 0$ can be done similarly. The lemma states that the points $(x_1^{\mathrm{opt}}, y_1), \ldots, (x_n^{\mathrm{opt}}, y_n)$ lie either on the regression line $y = \hat{\beta}_0^{\mathrm{opt}} + \hat{\beta}_1^{\mathrm{opt}} x$, or on the y-axis. Suppose there exist points $(x_i^{\mathrm{opt}}, y_i)$ to the contrary, that is, $x_i^{\mathrm{opt}} > 0$ and the point is not on the regression line. If $y_i < \hat{\beta}_0^{\mathrm{opt}}$, then since $\hat{\beta}_1^{\mathrm{opt}} > 0$, considering point $(0, y_i)$ instead of $(x_i^{\mathrm{opt}}, y_i)$ would give a smaller residual, contradicting the definition of $\boldsymbol{x}^{\mathrm{opt}}$. If $y_i > \hat{\beta}_0^{\mathrm{opt}}$, then again since $\hat{\beta}_1^{\mathrm{opt}} > 0$, $(x_i^{\mathrm{opt}}, y_i)$ cannot lie to the right of the regression line, or we can replace $(x_i^{\mathrm{opt}}, y_i)$ with a point $((y_i - \hat{\beta}_0^{\mathrm{opt}}) / \hat{\beta}_1^{\mathrm{opt}}, y_i)$ on the regression line with a smaller residual. We can also say that the points cannot lie *left* of the regression line, since we can construct a new regression line passing the point $(0, \beta_0)$ that lies left of the points, and replace all points $(x_i^{\mathrm{opt}}, y_i)$ with points on the new regression line. The residual of the new points are clearly smaller, and again contradicts the definition of $\boldsymbol{x}^{\mathrm{opt}}$. □

Corollary 1. *Let* $X^{\mathrm{opt}} = \{i \mid x_i^{\mathrm{opt}} = 0\}$. *If* $X^{\mathrm{opt}} = \emptyset$, $RSS(\boldsymbol{y}|\boldsymbol{x}^{\mathrm{opt}}) = 0$. *If* $X^{\mathrm{opt}} \neq \emptyset$, *then*

$$\hat{\beta}_0^{\mathrm{opt}} = \sum_{i \in X} y_i / |X^{\mathrm{opt}}|$$

and

$$RSS(\boldsymbol{y}|\boldsymbol{x}^{\mathrm{opt}}) = \sum_{i \in X^{\mathrm{opt}}} (\hat{\beta}_0^{\mathrm{opt}} - x_i^{\mathrm{opt}})^2.$$

Proof. As a consequence of Lemma 1. If $X = \emptyset$, then all points $(x_i^{\mathrm{opt}}, y_i)$ lie on the regression line. Otherwise, since the residual for all points $x_i^{\mathrm{opt}} > 0$ is 0, β_0 is chosen to minimize the residual sum of squares of points where $x_i^{\mathrm{opt}} = 0$. □

In order to calculate the lower bound, the problem now is how to obtain the value $\hat{\beta}_0^{\mathrm{opt}}$. Although the exact value of $\hat{\beta}_0^{\mathrm{opt}}$ is not known beforehand, Algorithm 2 shows how to calculate the minimum residual sum of squares for any $\boldsymbol{z} = (z_1, \ldots, z_n)^T$ satisfying the constraint: $0 \leq z_i \leq x_i$ for all $i = 1, \ldots, n$.

Corollary 2. *Let*

$$X_+^{\mathrm{opt}} = \{i \in X^{\mathrm{opt}} \mid y_i > \hat{\beta}_0^{\mathrm{opt}}\}$$
$$X_-^{\mathrm{opt}} = \{i \in X^{\mathrm{opt}} \mid y_i \leq \hat{\beta}_0^{\mathrm{opt}}\}.$$

If $\hat{\beta}_1^{\mathrm{opt}} > 0$, *then*

$$X_+^{\mathrm{opt}} = \{i \mid x_i = 0, y_i > \hat{\beta}_0^{\mathrm{opt}}\}$$
$$X_-^{\mathrm{opt}} = \{i \mid y_i \leq \hat{\beta}_0^{\mathrm{opt}}\}.$$

If $\hat{\beta}_1^{\mathrm{opt}} < 0$, *then*

$$X_+^{\mathrm{opt}} = \{i \mid x_i = 0, y_i < \hat{\beta}_0^{\mathrm{opt}}\}$$
$$X_-^{\mathrm{opt}} = \{i \mid y_i \geq \hat{\beta}_0^{\mathrm{opt}}\}.$$

Proof. As a consequence of Lemma 1. When $\hat{\beta}_1^{\mathrm{opt}} > 0$ and $y_i > \hat{\beta}_0^{\mathrm{opt}}$, $x_i^{\mathrm{opt}} = 0$ if and only if $x_i = 0$. Similarly, when $\hat{\beta}_1^{\mathrm{opt}} < 0$ and $y_i < \hat{\beta}_0^{\mathrm{opt}}$, $x_i^{\mathrm{opt}} = 0$ if and only if $x_i = 0$. □

Theorem 2. *Algorithm 2 correctly outputs* $RSS(\boldsymbol{y}|\boldsymbol{x}^{\mathrm{opt}})$.

Proof. Consider the case where $\{i \mid x_i^{\mathrm{opt}} = 0\} \neq \emptyset$ and $\hat{\beta}_1^{\mathrm{opt}} > 0$. The claim is that $\mathsf{m} = \hat{\beta}_0^{\mathrm{opt}}$, and $X = X^{\mathrm{opt}}$, after the **while** loop of lines 5–9. If this can be proved, the result follows from Corollary 1. Let us split the set X thus

Algorithm 2: Algorithm for calculating the lower bound of RSS.

```
   Input: $\boldsymbol{x} = (x_1, \ldots, x_n)^T$, $\boldsymbol{y} = (y_1, \ldots, y_n)^T$
   Output: $\min\{RSS(\boldsymbol{y}|\boldsymbol{z}) \mid \boldsymbol{z} = (z_1, \ldots, z_n)^T, 0 \le x'_i \le x_i\}$
1  $X \leftarrow \{i \mid x_i = 0\}$; $k \leftarrow 1$;
2  if $X = \emptyset$ then return $0$;        /* Can move all $x_i$ onto a single line. */
   // assuming $\hat{\beta}_1^{\mathrm{opt}} > 0$ ////////////////////////////////////////
3  $\mathsf{m} \leftarrow \sum_{i \in X} y_i / |X|$;                /* mean of values $\{y_i \mid x_i = 0\}$ */
4  $(i_1, \ldots, i_{n-|X|}) \leftarrow$ indices sorted in increasing order of $\{y_i \mid i \notin X\}$ ;
5  while $y_{i_k} \le \mathsf{m}$ do
6      $X \leftarrow X \cup \{i_k\}$;                                   /* $x_{i_k} \to 0$ */
7      $\mathsf{m} \leftarrow \sum_{i \in X} y_i / |X|$;                            /* update mean */
8      $k \leftarrow k + 1$;
9  endw
10 $\mathsf{lb} = \sum_{i \in X} (\mathsf{m} - x_i)^2$;        /* $\mathsf{m} = \hat{\beta}_0^{\mathrm{opt}}$ and $X = X^{\mathrm{opt}}$ if $\hat{\beta}_1^{\mathrm{opt}} > 0$ */
   // assuming $\hat{\beta}_1^{\mathrm{opt}} < 0$ ////////////////////////////////////////
11 $X \leftarrow \{i \mid x_i = 0\}$; $k \leftarrow 1$;
12 $\mathsf{m} \leftarrow \sum_{i \in X} y_i / |X|$;                /* mean of values $\{y_i \mid x_i = 0\}$ */
13 $(i_1, \ldots, i_{n-|X|}) \leftarrow$ indices sorted in decreasing order of $\{y_i \mid i \notin X\}$ ;
14 while $y_{i_k} \ge \mathsf{m}$ do
15     $X \leftarrow X \cup \{i_k\}$;                                   /* $x_{i_k} \to 0$ */
16     $\mathsf{m} \leftarrow \sum_{i \in X} y_i / |X|$;                            /* update mean */
17     $k \leftarrow k + 1$;
18 endw
19 return $\min\{\mathsf{lb}, \sum_{i \in X} (\mathsf{m} - x_i)^2\}$;   /* $\mathsf{m} = \hat{\beta}_0^{\mathrm{opt}}$ and $X = X^{\mathrm{opt}}$ if $\hat{\beta}_1^{\mathrm{opt}} < 0$ */
```

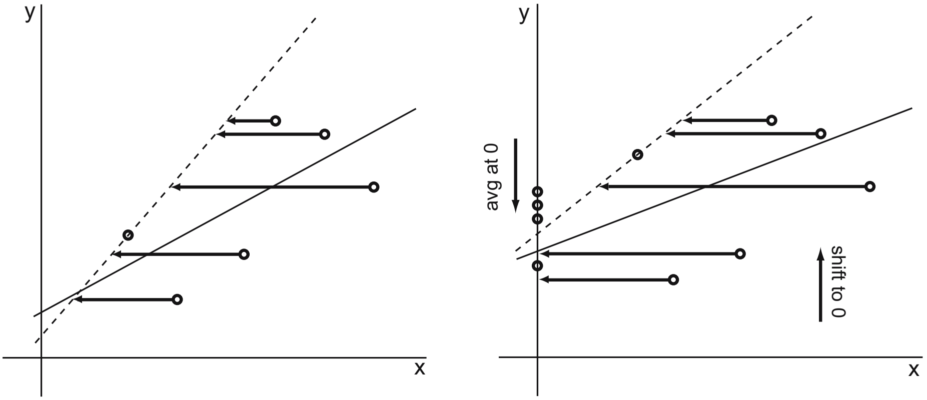


Fig. 2. Case of Algorithm 2 where $X = \emptyset$ (left) and $X \neq \emptyset$, $\beta_1 > 0$ (right)

calculated into two disjoint sets, $X_+ = \{i \in X \mid y_i > \mathsf{m}\}$ and $X_- = \{i \in X \mid y_i \le \mathsf{m}\}$. Since the algorithm does not add indices i where $y_i > \mathsf{m}$ to X, we have that $X_+ = \{i \mid x_i = 0, y_i > \mathsf{m}\}$ from the initial construction of X at line 1. Also, since all indices i where $y_i \le \mathsf{m}$ are added to X, we have that $X_- = \{i \mid y_i \leq \mathsf{m}\}$. Suppose $\mathsf{m} < \hat{\beta}_0^{\mathrm{opt}}$. From Corollary 2, we have that $X_+ \supseteq X_+^{\mathrm{opt}}$, and $X_- \subseteq \{i \mid y_i \leq \hat{\beta}_0^{\mathrm{opt}}\} = X_-^{\mathrm{opt}}$. However, this contradicts the assumption, since $\sum_{i \in X} y_i/|X| = \mathsf{m} \geq \hat{\beta}_0^{\mathrm{opt}} = \sum_{i \in X^{\mathrm{opt}}} y_i/|X^{\mathrm{opt}}|$. Next, suppose $\mathsf{m} > \hat{\beta}_0^{\mathrm{opt}}$. We have the opposite situation and $X_+ \subseteq X_+^{\mathrm{opt}}$ and $X_- \supseteq X_-^{\mathrm{opt}}$. However, due to the way X is constructed, there must have been a point in the while loop (lines 5–9) where all indices i where $y_i \leq \hat{\beta}_0^{\mathrm{opt}}$ are added to X, and no index i where $y_i > \hat{\beta}_0^{\mathrm{opt}}$ is added to X. From Corollary 2, at such point, $X_+ = X_+^{\mathrm{opt}}$ and $X_- = X_-^{\mathrm{opt}}$, which implies that $\mathsf{m} = \hat{\beta}_0^{\mathrm{opt}}$. Due to the condition of the while loop, no other index i where $y_i > \hat{\beta}_0^{\mathrm{opt}}$ could have been added to X afterwards, contradicting the assumption $\mathsf{m} > \hat{\beta}_0^{\mathrm{opt}}$. Therefore, $\mathsf{m} = \hat{\beta}_0^{\mathrm{opt}}$, and consequently $X = X^{\mathrm{opt}}$. □

Fig. 2 shows the basic idea of Algorithm 2. The time complexities of Algorithm 1 and 2 are both $O(n)$, since they just conduct a constant number of scans on the data set, provided that the data is initially sorted and ranked according to y_i, so that the sorting at lines 4 and 13 of Algorithm 2 can be computed in $O(n)$ time.

Combining Suffix Tree Traversal and Branch and Bound. It is easy to see that for substring patterns, the generalized suffix tree itself can be used as the search tree, and we can combine the $O(N^2)$ algorithm and the pruning strategy described in this section. Combining the $O(Nn)$ algorithm and the pruning strategy is not readily realizable, since the direction of the traversal over the search tree is in the *reverse* direction. We discuss this issue further in Section 5.

4 Computational Experiments

We implement our algorithms using the C++ language, and measure the running times of our algorithm using a Sun Fire 15K (UltraSPARC III Cu 1.2GHz x 96 CPUs) using a single processor for each run. We note that the suffix tree algorithm is simulated by a suffix array structure, using the method presented in [17,18].

For the numeric attribute values, we use the *S. cerevisiae* gene expression data obtained from microarray experiments given in [19]. The data consists of normalized log expression level ratios of genes at specific time points of the yeast cell cycle. For the string data, we use the 600 nucleotides from the upstream of the start codon of each gene.

Table 1 shows the running times of the algorithms for finding the optimal substring pattern applied to the expression data of the 14-minute time point in the α-synchronized cell-cycle microarray experiment. The times are measured for various sizes of n by a random sampling from the entire set of 5907 genes which were available for this time point. Note that since all strings are of fixed length, $N = 600n$. From the table, we see that the $O(Nn)$ and $O(N^2)$ time algorithms are able to find the optimal pattern in a reasonable amount of time. However, we

Table 1. Comparison of running times of algorithm for finding optimal substring patterns from the 14-minute time point in the α-synchronized cell-cycle microarray experiment [19].

n	$O(Nn)$	$O(N^2)$	$O(N^2)$+ simple bb	$O(N^2)$+bb
100	05.96s	12.43s	05.90s	05.71s
500	00m37s	01m47s	00m30s	00m26s
1000	02m03s	05m28s	01m02s	00m53s
1500	04m09s	11m16s	01m35s	01m19s
2000	07m53s	19m55s	02m00s	01m38s
2500	12m30s	30m05s	02m35s	02m07s
3000	18m20s	42m17s	03m29s	02m51s
3500	25m17s	56m34s	04m11s	03m23s
4000	33m08s	73m06s	04m52s	03m55s
4500	42m48s	92m16s	05m34s	04m28s
5000	55m03s	113m20s	06m13s	04m59s
5500	67m01s	136m25s	07m08s	05m41s
5907	75m55s	157m27s	07m57s	06m21s

can see that $O(N^2)$+bb (the $O(N^2)$ algorithm with the branch and bound strategy Algorithm 2) is much faster for all input sizes. Although $O(N^2)$+simple bb (the $O(N^2)$ algorithm with the simple branch and bound strategy Algorithm 1) is fairly efficient as well, the extra work invested in the $O(N^2) +$ bb algorithm is payed off by a $\sim 20\%$ reduction in the overall computation time.

To show that the algorithm also allows for the discovery of more complex patterns in a practical amount of time, we searched for the optimal don't care pattern on the same data set. The search took 765 minutes and 529 minutes, respectively, using simple bb and bb with a simple enumeration of don't care patterns, limiting the maximum length of the pattern to 15 and the number of don't cares characters in the pattern to 20% of its length.

5 Discussion

We presented efficient algorithmic solutions to the problem of discovering the optimal pattern in terms of a linear least squares fitting of the numeric attribute values associated with strings, and the matching function values. The efficiency of the algorithms are confirmed through computational experiments conducted on actual biological data.

In [20], the branch-and-bound enumerative search was applied in the *reverse* direction for finding the optimal degenerate pattern that discriminates between a positive string set and negative string set, where the matching function is an indicator function. In their search, the search tree is essentially traversed bottom-up. A bound on the score is computed in a similar way as for the original direction, and the traversal on the search can be pruned. This reverse direction is, however, difficult to achieve for the problem considered in this paper. This is because in the original direction, the bound is calculated from the numeric attribute values of strings which do not match the pattern ($\psi_p(s_i) = 0$), which is possible due to the condition $\psi_p(s_i) \geq 0$. In order to calculate a bound for the reverse direction, we would need to assume a maximum value c where $\psi_p(s_i) \leq c$, and we would calculate a bound for the residual sum of squares from the numeric attribute values of strings which give $\psi_p(s_i) = c$. However, the matching function used in this paper is not suitable, since the maximum value would change for each string.

The lower bound calculated in this paper underestimates the actual minimum value that can be achieved with the matching function and numeric attribute values. This is because we did not require that there exist a pattern whose matching function values would be equal to $\boldsymbol{x}^{\mathrm{opt}}$. Notice that when calculating the lower bound, $\boldsymbol{x}^{\mathrm{opt}}$ is obtained by considering points on the regression line. However, we know that the matching function will only take discrete integer values, and it may not be possible for some x_i^{opt} to lie on the regression line. Residuals for such points will not be zero, and would therefore increase the lower bound. Finding an *efficient* way to calculate a better lower bound for the discretized version of the problem is an interesting open problem.

Acknowledgements

Computation time was provided in part by the Super Computer System, Human Genome Center, Institute of Medical Science, University of Tokyo.

References

1. Brazma, A., Jonassen, I., Eidhammer, I., Gilbert, D.: Approaches to the automatic discovery of patterns in biosequences. J. Comput. Biol. **5** (1998) 279–305
2. Hirao, M., Hoshino, H., Shinohara, A., Takeda, M., Arikawa, S.: A practical algorithm to find the best subsequence patterns. Theoretical Computer Science **292** (2002) 465–479
3. Shinohara, A., Takeda, M., Arikawa, S., Hirao, M., Hoshino, H., Inenaga, S.: Finding best patterns practically. In: Progress in Discovery Science. Volume 2281 of LNAI., Springer-Verlag (2002) 307–317
4. Takeda, M., Inenaga, S., Bannai, H., Shinohara, A., Arikawa, S.: Discovering most classificatory patterns for very expressive pattern classes. In: 6th International Conference on Discovery Science (DS 2003). Volume 2843 of LNCS., Springer-Verlag (2003) 486–493
5. Hirao, M., Inenaga, S., Shinohara, A., Takeda, M., Arikawa, S.: A practical algorithm to find the best episode patterns. In: Proc. 4th International Conference on Discovery Science (DS2001). Volume 2226 of LNAI., Springer-Verlag (2001) 435–440
6. Inenaga, S., Bannai, H., Shinohara, A., Takeda, M., Arikawa, S.: Discovering best variable-length-don't-care patterns. In: Proceedings of the 5th International Conference on Discovery Science. Volume 2534 of LNAI., Springer-Verlag (2002) 86–97

7. Bussemaker, H.J., Li, H., Siggia, E.D.: Regulatory element detection using correlation with expression. Nature Genetics **27** (2001) 167–171
8. Bannai, H., Inenaga, S., Shinohara, A., Takeda, M., Miyano, S.: A string pattern regression algorithm and its application to pattern discovery in long introns. Genome Informatics **13** (2002) 3–11
9. Bannai, H., Inenaga, S., Shinohara, A., Takeda, M., Miyano, S.: Efficiently finding regulatory elements using correlation with gene expression. Journal of Bioinformatics and Computational Biology **2** (2004) 273–288
10. Zilberstein, C.B.Z., Eskin, E., Yakhini, Z.: Using expression data to discover RNA and DNA regulatory sequence motifs. In: The First Annual RECOMB Satellite Workshop on Regulatory Genomics. (2004)
11. Bannai, H., Hyyrö, H., Shinohara, A., Takeda, M., Nakai, K., Miyano, S.: An $O(N^2)$ algorithm for discovering optimal Boolean pattern pairs. IEEE/ACM Transactions on Computational Biology and Bioinformatics **1** (2004) 159–170 (special issue for selected papers of WABI 2004).
12. Hui, L.: Color set size problem with applications to string matching. In: Proceedings of the Third Annual Symposium on Combinatorial Pattern Matching (CPM 92). Volume 644 of LNCS., Springer-Verlag (1992) 230–243
13. Miyano, S., Shinohara, A., Shinohara, T.: Which classes of elementary formal systems are polynomial-time learnable? In: Proceedings of the 2nd Workshop on Algorithmic Learning Theory. (1991) 139–150
14. Miyano, S., Shinohara, A., Shinohara, T.: Polynomial-time learning of elementary formal systems. New Generation Computing **18** (2000) 217–242
15. Gusfield, D.: Algorithms on Strings, Trees, and Sequences. Cambridge University Press (1997)
16. Knuth, D.E., Morris, J.H., Pratt, V.R.: Fast pattern matching in strings. SIAM Journal on Computing **6** (1977) 323–350
17. Kasai, T., Lee, G., Arimura, H., Arikawa, S., Park, K.: Linear-time longest-common-prefix computation in suffix arrays and its applications. In: 12th Annual Symposium on Combinatorial Pattern Matching (CPM 2001). Volume 2089 of LNCS., Springer-Verlag (2001) 181–192
18. Kasai, T., Arimura, H., Arikawa, S.: Efficient substring traversal with suffix arrays. Technical Report 185, Department of Informatics, Kyushu University (2001)
19. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. Mol. Biol. Cell **9** (1998) 3273–3297
20. Shinozaki, D., Akutsu, T., Maruyama, O.: Finding optimal degenerate patterns in DNA sequences. Bioinformatics **19** (2003) ii206–ii214

Named Entity Recognition for the Indonesian Language: Combining Contextual, Morphological and Part-of-Speech Features into a Knowledge Engineering Approach

Indra Budi[1], Stéphane Bressan[2], Gatot Wahyudi[1], Zainal A. Hasibuan[1],
and Bobby A.A. Nazief [1]

[1] Faculty of Computer Science University of Indonesia
{indra, zhasibua, nazief}@cs.ui.ac.id, gatot100@mhs.cs.ui.ac.id
[2] School of Computing, National University of Singapore
steph@nus.edu.sg

Abstract - We present a novel named entity recognition approach for the Indonesian language. We call the new method InNER for *Indonesian Named Entity Recognition*. InNER is based on a set of rules capturing the contextual, morphological, and part of speech knowledge necessary in the process of recognizing named entities in Indonesian texts. The InNER strategy is one of knowledge engineering: the domain and language specific rules are designed by expert knowledge engineers. After showing in our previous work that mined association rules can effectively recognize named entities and outperform maximum entropy methods, we needed to evaluate the potential for improvement to the rule based approach when expert crafted knowledge is used. The results are conclusive: the InNER method yields recall and precision of up to 63.43% and 71.84%, respectively. Thus, it significantly outperforms not only maximum entropy methods but also the association rule based method we had previously designed.

1 Introduction

Named entity recognition is the task of recognizing and classifying terms and phrases as named entity from free text [13]. It is a fundamental building block of many, if not all, textual information extraction strategies. It is going to be a crucial component of most tools for the construction of a semantic Web layer on top of the existing wealth of textual information available on the World Wide Web. We could also applied information extraction into scientific documents. Some interesting entities could be extracted from the document. For example, we could extract the statement that indicates problem, objective, method and result of the research from the abstract of the document.

Typically useful named entity classes are names of person, locations, organizations, money amounts, percentages and dates. Named entity recognition is the first step towards the extraction of structured information from unstructured texts. For example in the following text in the Indonesian language we can recognize `*Habibie'* and `*Amien Rais'* as names of person and `*Jakarta'* as a location.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 57 – 69, 2005.

Presiden Habibie bertemu dengan Prof. Amien Rais di Jakarta kemarin.
(President Habibie met Prof. Amien Rais in Jakarta yesterday)

The recognition task is usually leveraging features of the terms such as their morphology, part of speech, and their classification and associations in thesauri and dictionaries. It is also leveraging the context in which terms are found such as neighboring terms and structural elements of the syntactical units, for instances propositions, sentences, and paragraphs.

Clearly the characteristic combinations of the above features differ significantly from one language to another. Techniques developed for the American language need to be adapted to non-English linguistic peculiarities. It is also possible that entirely new and specific techniques need to be designed.

This research is part of a larger project aiming at the design and development of tools and techniques for the Indonesian Web.

We here present a novel named entity recognition approach for the Indonesian language. We call the new method InNER for *Indonesian Named Entity Recognition.* InNER is based on a set of rules capturing the contextual, morphological, and part of speech knowledge necessary in the process of recognizing named entities in Indonesian texts.

The InNER strategy is one of knowledge engineering: the domain and language specific rules are designed by an expert knowledge engineer [1]. After showing in our previous work [6] that mined association rules can effectively recognize named entities and outperform maximum entropy methods, we now need to evaluate the potential for improvement to the rule based approach when expert crafted knowledge is added.

The rest of this paper organized as follows. We present and discuss some background and related works on named entity recognition in the next section. In the third section, we present the InNER strategy and its implementation. We then present the results of an evaluation of its performance in section 4. We empirically evaluate the effectiveness of the InNER method and compare it with one of methods that we had previously showed to outperform existing methods when applied to the Indonesian language. We present conclusions on section 5 and finally, we give an outline of the directions for future work in section 6.

2 Background and Related Works

There are two common families of approaches to named entity recognition, knowledge engineering approaches and machine learning approaches [1]. Knowledge engineering approaches are expert-crafted instances of generic models and techniques to recognize named entity in the text. Such approaches are typically rule-based. In a rule-based approach the expert design rules to be used by a generic inference engine. The rule syntax allows the expression of grammatical, morphological and contextual patterns. The rules can also include dictionary and thesauri references. For example, the following rule contributes to the recognition of persons.

If a proper noun is preceded by a title **then** the proper noun is name of person

In [2], the authors introduce the FASTUS system whose rules are using regular expressions. In [14], the authors built knowledge representation that consist on rules to identify name entities based on geographical knowledge, common person names and common organization names. In [16], the authors use a semantic network consisting of some 100.000 nodes and hold information such as concepts hierarchies, lexical information, and prototypical events. All the above works are applied to the English language.

In machine learning approaches, a generic computer program learns to recognize named entities with or without training and feedback. Very general machine learning models exist that do not necessitate the mobilization of expensive linguistic expert knowledge and resources. For instance the Nymble system [3] uses a hidden Markov model. In both [2] and [7], the authors present an approach that uses the now popular maximum entropy. These models can make use of different features. For instance, in [4], the authors use morphological features, paragraphs and a dictionary. In [7], the authors combine local and global features.

As mentioned in [10], the knowledge engineering and the machine learning approach are not incompatible. For instance, the system presented in [15] combines rules and maximum entropy to recognize named entity in English texts.

Our first attempt to design a named entity recognition system for the Indonesian language was of the machine learning family [6]. We mined a set of association rules from a training corpus in which we considered the sequences of terms annotated with their features and name class.

Numerous other authors have worked on named entity recognition for non-English languages. Some have made their results available. In [18], the authors propose a named entity recognition approach based on decision tree for the Japanese language. In [12], the authors proposed a rule based approach for financial texts in the Greek language. In [19], the authors use a combination of lexical, contextual and morphological feature to recognize named entities for the Turkish language. In [9], the authors present an approach combining rules with machine learning for the Swedish language.

3 Name Entity Recognition

The approach we propose in this paper is based on the expert engineering of rules for the recognition of named entities. The rules are designed and verified by educated native speakers after analysis of a training corpus. A rule combines contextual, morphological, and part of speech features to assign a class to terms and groups of terms in the text.

The class of a named entity can be directly recognized from its context. For example, in a sentence comprising a title particle such as “Prof.” followed by proper name, the proper name is the name of a person. For example, in the sentence: “Prof. Yusuf berkunjung ke Jakarta”. The term “Yusuf” is recognized as the name of a person because it is a proper name preceded by term which belongs to contextual information (“Prof.”).

In the above example, we can directly infer that the term ‘Yusuf’ is a proper name because of its spelling with an upper case in the beginning. The format and nature of the characters forming terms give some basic indications: lower and upper cases, signs, diacritics, and digits.

In the above example, the term ‘Yusuf’ belongs to the morphological proper name. It means ‘Yusuf’ is a proper name morphologically.

We assume the availability of the necessary tools for the lexical analysis and part of speech tagging of the text.

The knowledge engineering task for the expert is the design of rules identifying the chosen named entity classes based on contextual, morphological, and part of speech information as explained above. As we have seen in the introduction, an example rule reads as follows.

If a proper noun is preceded by a title **then** the proper noun is the name of person
If a proper noun is preceded by ‘di’ **then** the proper noun is the name of a location

The InNER system processes the input text sentence by sentence. To each input sentence, the corresponding output is a sentence in which the text corresponding to a recognized named entity is marked-up with XML tags following the widely used framework introduced in [8].

For instance, the processing of the example sentence “*Presiden Habibie bertemu dengan Prof. Amien Rais di Jakarta kemarin*” outputs the following XML tagged text:

```
Presiden <ENAMEX TYPE=”PERSON”>Habibie</ENAMEX> bertemu dengan Prof.
<ENAMEX TYPE=”PERSON”>Amien Rais</ENAMEX> di <ENAMEX
TYPE=”LOCATION”>Jakarta</ENAMEX> kemarin.
```

The InNER system has four main processes, as depicted on Fig. 1: tokenization, feature assignment, rule assignment and name tagging.

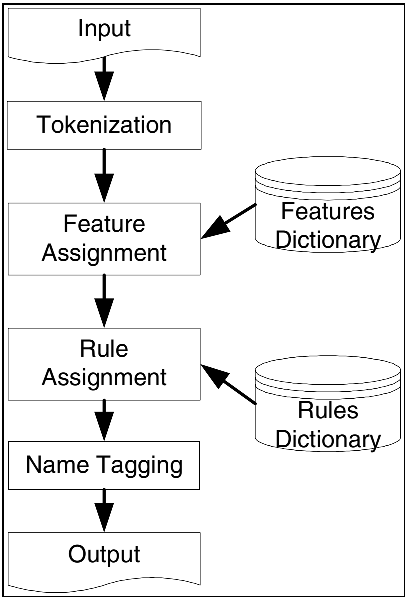


Fig. 1. The InNER architecture

The purpose of the tokenization process is to identify tokens (words, punctuation and other units of text such as numbers etc.) from the input sentence. Tokens are labeled with their kinds. For example, table 4; illustrate the results of the tokenization of the sentence below in which the tokenizer identifies words (WORD), punctuation (OPUNC, EPUNC) and numbers (NUM).

Ketua MPR, Amien Rais pergi ke Bandung kemarin (24/4).
(Chief of MPR, Amien Rais went to Bandung yesterday (24/4).)

Table 1. List of contextual features

Feature Name	Explanation	Example
PPRE	Person prefix	Dr., Pak, K.H.,
PMID	Person middle	bin, van
PSUF	Person suffix	SKom, SH
PTIT	Person title	Menristek, Mendagri
OPRE	Organization prefix	PT., Universitas
OSUF	Organization suffix	Ltd., Company
OPOS	Position in organization	Ketua
OCON	Other organization contextual	Muktamar, Rakernas
LPRE	Location prefix	Kota, Propinsi
LSUF	Location suffix	Utara, City
LLDR	Location leader	Gubernur, Walikota
POLP	Prepositions that's usually followed by person name	oleh, untuk
LOPP	Prepositions that's usually followed by location name	di, ke, dari
DAY	Day	Senin, Sabtu
MONTH	Month	April, Mei

Table 2. List of morphological features

Feature Name	Explanation	Example
TitleCase	Begin with uppercase letter and followed by all lowercase letter	Soedirman
UpperCase	All uppercase letter	KPU
LowerCase	All lowercase letter	menuntut
MixedCase	Uppercase and lowercase letter	LeIP
CapStart	Begin with uppercase letter	LeIP, Muhammad
CharDigit	Letter and number	P3K
Digit	All number	2004
DigitSlash	Number with slash	17/5
Numeric	Number with dot or comma	20,5; 17.500,00
NumStr	Number in word	satu, tujuh, lima
Roman	Roman number	VII, XI
TimeForm	Number in time format	17:05, 19.30

The feature assignment component labels the terms with their features, the basic contextual features (for instance identifying preposition, days, or titles), the morphological features, as well as the part of speech classes. The identification of contextual features uses the context dictionary. The analysis of the morphological features parses the token. The identification of the part of speech classes is a complex process. We use part of speech tagging technology developed by our team [5, 17].

Contextual feature example illustrated in the Table 1, while Table 2 and 3 illustrate and provide examples for morphological and part of speech features, respectively. For example, table 4; illustrate the result of feature assignment process on the above sentence.

Table 3. List of part-of-speech features

Feature Name	Explanation	Example
ART	Article	si, sang
ADJ	Adjective	indah, baik
ADV	Adverb	telah, kemarin
AUX	Auxiliary verb	harus
C	Conjunction	dan, atau, lalu
DEF	Definition	merupakan
NOUN	Noun	rumah, gedung
NOUNP	Personal noun	ayah, ibu
NUM	Number	satu, dua
MODAL	Modal	akan
OOV	Out of dictionary	
PAR	Particle	kah, pun
PREP	Preposition	di, ke, dari
PRO	Pronominal	saya, beliau
VACT	Active verb	menuduh
VPAS	Passive verb	dituduh
VERB	Verb	pergi, tidur

The rule assignment component selects the candidate rules for each identified token in the text. The actual orchestration and triggering of the rules occur in the name tagging component.

Table 4. Result of tokenization and feature assigment processes

Token string	Token kind	Contextual features	Morphological features	Part-of-speech features
Ketua	WORD	OPOS	TitleCase, CapStart	NOUN
MPR	WORD		UpperCase, CapStart	OOV
,	OPUNC			
Amien	WORD		TitleCase, CapStart	OOV
Rais	WORD		TitleCase, CapStart	Noun
pergi	WORD		LowerCase	VERB
ke	WORD		LowerCase	PREP
Bandung	WORD		TitleCase, CapStart	NOUN
kemarin	WORD		LowerCase	NOUN, ADV
(	SPUNC			
24/4	NUM		DigitSlash	
)	EPUNC			
.	OPUNC			

The rules in the InNER system capture the typical patterns of features characterizing the various named entity classes. The left hand side of a rule is the pattern. The right hand side of a rule is the identified named entity class. The following is an example of a rule.

```
IF     Token[i].Kind="WORD" and Token[i].OPOS and Token[i+1].Kind=”WORD” and
       Token[i+1].UpperCase and Token[i+1].OOV
THEN   Token[i+1].NE = "ORGANIZATION"
```

Rule above would recognize token “*MPR*” as an organization. Table 5 shows the result of the rule assignment process for the example sentence. The empty string indicates that the term or phrase has not been classified. In the example, “*MPR*”, the Indonesian parliament, is identified as an organization. “*Amien Rais*”, an Indonesian politician, is identified as a person. “*Bandung*”, an Indonesian city is identified as a location.

There is no such order of rules, we arrange the rules randomly without follow certain mechanism. We choose the first match between each token and the rules, then we applied the rules to that token.

Table 5. Result of rule assigment process

Token	Type of Named Entity
Ketua	“”
MPR	ORGANIZATION
,	“”
Amien	PERSON
Rais	PERSON
Pergi	“”
Ke	“”
Bandung	LOCATION
Kemarin	“”
(	“”
24/4	“”
)	“”

The last process in InNER system is the XML tagging of the original sentence. The name tagging arranges some tokens that are identified as same class and position consecutively in the text into one single name entity class. The syntax of the tags follows mechanism in [8]. The example below is the output of the system for our running example. Term “Amien” and “Rais” are positioned consecutively and identified as same class PERSON, tagged together as one single name entity class (PERSON).

```
Ketua <ENAMEX TYPE="ORGANIZATION">MPR</ENAMEX>, <ENAMEX
TYPE=”PERSON”>Amien Rais</ENAMEX> pergi ke <ENAMEX
TYPE=”LOCATION”>Bandung</ENAMEX> kemarin (24/4).
```

4 Performance Analysis

We now empirically analyze the performance of the newly proposed approach. This analysis is done in comparison with our previous association rule based approach. We recall that we have shown in [6] that the latter outperforms existing methods (maximum entropy) for the named entity recognition task for the Indonesian language.

4.1 Experimental Setup and Metrics

For this evaluation we consider three named entity classes: names of person, locations and organizations. Our experts are graduate students who are native speakers of Indonesian language.

The observation corpus is composed of a set of articles from the online versions of two Indonesian newspaper Kompas (www.kompas.com) and Republika (www.republika.co.id). The observation corpus consists of 802 sentences. It comprises 559 names of person, 853 names of organization, and 418 names of location. Our testing corpus consists of 1.258 articles from the same sources. It includes 801 names of person, 1.031 names of organization, and 297 names of location. Both the observation and testing corpora have been independently tagged by native speakers based on guideline provided in [20].

We wish to measure the effectiveness of the approaches empirically evaluated. For this we use the definitions of the recall, precision and F-Measure metrics proposed by MUC (Message Understanding Conference) in [11].

These definitions use the following measurements.

- *Correct*: number of correct recognition performed by the system
- *Partial*: number of partial correct recognition performed by the system. for example:
  the phrase "*Amien Rais* " should be recognize as a person but the system just recognize "*Amien*" as PERSON or just recognize "*Rais*" as a PERSON.
- *Possible*: number named entity in the text as manually tagged for the training.
- *Actual*: number of tagged named entity output by the system. They may be correct, partially correct, or incorrect (we call incorrect tagged terms which are neither correct nor partially correct)

Based on the values above, the system performance parameters can be calculated in term of recall, precision and F-Measure using following formula.

$$Recall = \frac{Correct + 0.5 * Partial}{Possible} \tag{1}$$

$$Precision = \frac{Correct + 0.5 * Partial}{Actual} \tag{2}$$

$$F - Measure = \frac{Recall * Precision}{0.5 * (Recall + Precision)} \tag{3}$$

Let us illustrate the above definition with our example sentence manually tagged as given in section 1:

Presiden <ENAMEX TYPE="PERSON">Habibie</ENAMEX> bertemu dengan Prof. <ENAMEX

TYPE="PERSON">Amien Rais</ENAMEX> di <ENAMEX TYPE="LOCATION">Jakarta
</ENAMEX> kemarin.

Namely, there are three named entities: Habibie, Amien Rais and Jakarta of respective class person, person and location. Let us now assume that the same sentence is tagged by the system as follows.

Presiden <ENAMEX TYPE="PERSON">Habibie</ENAMEX> bertemu dengan Prof.
<ENAMEX
TYPE="PERSON">Amien</ENAMEX> <ENAMEX TYPE="ORGANIZATION">Rais</ENAMEX>
di<ENAMEX TYPE="LOCATION">Jakarta</ENAMEX> kemarin.

Namely, the system identifies four named entities: Habibie, Amien, Rais and Jakarta of respective class person, person, organization and location. The first and fourth terms are correctly tagged. The second term ‘Amien’ is partially tagged as it should be ‘Amien Rais’. The third term ‘Rais’ is wrongly tagged. Therefore recall, precision and F-Measure for this sentence alone are computed as follows.

Recall = (2 + 0.5)/3 = 2.5/3 = 83.33%
Precision = (2 + 0.5)/4 = 2.5/4 = 62.50%
F-Measure = 71.43%

4.2 Results and Analysis

Our experts engineered a total of 100 rules by examined observation document. We have classified the rules in four categories depending on the combination of features that they are using.

We call contextual rules, rules involving contextual features only. This is the base set of rules. There are 18 such rules.

We call CM rules, rules that combine contextual and lexical features. There are 33 such rules.

We call CP rules, rules that combine contextual and part of speech features. There are 27 such rules.

We call CMP rules, rules that combine all features. There are 22 such rules.

Table 6. Result on different combination of the rule sets

Rules	Recall	Precision	F-Measure
Contextual (Base)	35.79%	33.87%	34.82%
Base + CP	46.81%	49.80%	48.26%
Base + CM	47.91%	70.30%	56.98%
Base + CP + CM + CMP	63.43%	71.84%	67.37%

Table 6 shows the performance of different combinations of the rule sets. Surprisingly morphological features seem to yield better results than part of speech features. This is probably due to the named entity classes that we are considering for which upper case first character is often a determinant indicator. However, regardless of the specificity of our choice of named entity classes and as we expected, the best overall results for recall, precision and F-measure are obtained from the combination of all types of rules and all types of features. Using all combinations of all rules is the strategy we propose and call the InNER strategy.

For further language processing steps, the partial correct is not useful, so we also give result without partial correct, only calculate the correct response. Table 7 shows the result of system without partially correct and table 8 give the difference on F-Measure between system with partial correct and without partial correct.

Table 7. Result of experiment without partial correct

Rules	Recall	Precision	F-Measure
Contextual (Base)	28.28%	26.74%	27.49%
Base + CP	41.29%	43.93%	42.57%
Base + CM	44.11%	64.71%	52.49%
Base + CP + CM + CMP	60.22%	67.76%	63.77%

Table 8. Difference between with and without partial correct on F-Measure

Rules	With Partial Correct	Without Partial Correct	Difference
Contextual (Base)	34,82%	27,49%	7,33%
Base + CP	48,26%	42,57%	5,69%
Base + CM	56,98%	52,49%	4,49%
Base + CP + CM + CMP	67,37%	63,77%	3,60%

Based on table 8, we saw that the contextual feature give highest contribution of partial correct and the all combination of features give lowest difference result when using partial correct. It means that adding more features give more accurate result.

We could see that the result are still below standard performance as compare to some other languages for NER (e.g. 80%). We have no definitive answer yet why this is happen since this is the first generation of NER in Indonesian language. It maybe because of the datasets used and decision to choose the rules. We develop the datasets manually and carefully, but we have no judgment from the domain expert about correctness of the datasets. Moreover, if we look manually at the datasets, we find that so many occurrences that a conjuction "***dan***" (and) used as part of organization name, for example: "*Fakultas Ilmu Sosial* ***dan*** *Politik" (Political and Social Faculty)*, "*Departemen Kehakiman* ***dan*** *HAM"* (*Human Right and Court Department*), "*Pusat Studi Demokrasi* ***dan*** *HAM" (Center of Human Right and Democracy Study)*, etc. Our system could detect those terms as two entities instead of one entity. The decision of choosing the right rules could be contribute to the lower result. Maybe if we could design the ranked of rules, the result would be better.

When comparing the performance of the InNER strategy, i.e. for all rules, to the performance of a named entity recognition by means of mined association rules (which we had shown in [6] outperforms maximum entropy methods for the Indonesian language) we find that InNER yields a consistently and significantly better performance in both recall and precision and therefore, naturally, in F-measure. Even though if we compare it to the combination of contextual and morphological rules, the InNER still have better performance.

We used observation document as training set to discover those association rules. We applied feature rule as association rules which used in [6] that form:

$<t_1, f_2> \Rightarrow nc_2$, *(support, confidence)*

Rule above was constructed from a sequence of terms <t_1, t_2>, where f_2 is the morphological feature of t_2 and nc_2 is the name class of t_2. The left hand side of the association rules are sequences of terms and features while the right hand side is the name class. The support and confidence depend on the occurence this form in the training sets. See [6] for further detail how this association rules could be used in name entity recognition. This form similar to the third rule of InNER, combination contextual and morphological feature.

Table 9 contains the figures of this comparison.

Table 9. Comparation with association rules method

Method	Recall	Precision	F-Measure
InNER	63.43%	71.84%	67.37%
Association Rules	43.33%	52.50%	47.49%

A manual closer look at the results, going through the correct, partial, possible, and actual named entities, seems to indicate that association rules induce more partial recognition. This fact also show that the performance of association rules closer enough with performance InNER using combination of lexical and morphological feature without partially correct. This is avoided by the knowledge engineering approach, which is capable of a finer grain tuning of the rule by leveraging the variety of features available.

5 Conclusions

We have proposed a knowledge engineering approach to the recognition of named entities in texts in the Indonesian language. The approach is based on rules that combine contextual, morphological, and part of speech features in the recognition process.

The method yields a highest performance of 63.43% recall and 71.84% precision with combine all of three features. Based on experiment, we also showed that morphological feature have better result than part-of-speech feature, it means that knowing the structure of letter forming a term give better result rather than its part-of-speech.

We showed that this method outperforms an association rule based method we had previously developed because this method reduce the partially correct result. Since we had previously shown, under a similar experimental set up, that the association rule based yielded a better performance than state of the art methods (maximum entropy) [6], we can conclude that based on our experiment, the knowledge engineering method is the best.

6 Future Work

Clearly this comes at the cost and expenses of expert knowledge and effort. Our experts have manually designed 100 rules. It is a tedious task, which we did not conduct, to compare these rules individually with the association rules that are automatically mined. It would be however interesting to compare and integrate the mined association rules and the engineered rules. Indeed, not only do we expect the controlled merging of the mined association rules with the engineered rules to results in an even more efficient method, but also we do expect an effective and elegant tool for visualizing and browsing the mined association rules to help the knowledge engineer in the design process itself.

For better performance, specifically to get the standard performance, we think there is should be improvement on the datasets beside the method. The datasets should be evaluated and revised by the domain expert in order to reducing the manual error.

The next step in our project is to devise a method to reconstruct structured elements from the elementary name entities identified. Our target language is XML. To illustrate our idea, let us consider the motivating example from which we wish to extract an XML document describing the meeting taking place:

"Presiden Habibie bertemu dengan Prof. Amien Rais di Jakarta kemarin."

Fig. 2 contains the manually constructed XML we hope to obtain. In italic are highlighted the components that require global, ancillary, or external knowledge. Indeed, although, we expect similar methods (rules based, association rules) can be applied to learn the model of combination of elementary entities into complex elements, we also expect that global, ancillary, and external knowledge will be necessary such as gazetteers (Jakarta is in Indonesia), document temporal and geographical context (Jakarta, 05/06/2003), etc.

```
<meeting>
  <date>05/06/2003</date>
  <location>
    <city>Jakarta</city>
    <country>Indonesia</country>
  </location>
  <participants>
    <person>
      <name>Habibie</name>
      <quality>Presiden</quality>
    </person>
    <person>
      <name>Amien Rais</name>
      <quality>Prof.</quality>
   </person>
  </participants>
  </meeting>
```

Fig. 2. Extracted structural form in XML

References

[1] Appelt, D., and Israel, D.J.: *Introduction to Information Extraction Technology*, Tutorial at IJCAI-99, Stockholm, Sweden (1999)

[2] Appelt, D., et al.: *SRI International FASTUS system MUC-6 test results and analysis*, In Proceedings of the 6th Message Understanding Conference (MUC-6) (1995)

[3] Bikel, D., et al.: *NYMBLE: A High Performance Learning Name-Finder*, In Proceeding of the fifth Conference on Applied Natural Language Processing, pp 194-201 (1997)

[4] Borthwick, A., et al.: *Exploiting diverse knowledge sources via maximum entropy in named entity recognition*. In Proceedings of the Sixth Workshop on Very Large Corpora, Montreal, Canada (1998)

[5] Bressan, S., and Indradjaja, L.: *Part-of-Speech Tagging without Training*, in Proc. of Intelligence in Communication Systems: IFIP International Conference, LNCS V3283, Springer-Verlag Heidelberg, ISSN: 0302-9743, ISBN: 3-540-23893-X (INTELLCOMM) (2004)

[6] Budi, I., and Bressan, S.: *Association Rules Mining for Name Entity Recognition*, In Proceeding of 4th Web Information System Engineering (WISE) Conference, Roma (2003)

[7] Chieu, H.L., and Ng, H.T.: *Named Entity Recognition: A Maximum Entropy Approach Using Global Information*, In Proceedings of the 19th International Conference on Computational Linguistics (2002)

[8] Chinchor, N., et al.: *Named Entity Recognition Task Definition Version 1.4*, The MITRE Corporation and SAIC (1999)

[9] Dalianis, H., and Åström, E.: *SweNam-A Swedish Named Entity recognizer. Its construction, training and evaluation*, Technical report, TRITA-NA-P0113, IPLab-189, NADA, KTH (2001)

[10] Dekang, L.: *Using Collocation Statistics in Information Extraction*, In Proceedings of the 7th Message Understanding Conference (MUC-7) (1998)

[11] Douthat, A.: *The Message Understanding Conference Scoring Software User's Manual*, In Proceedings of the 7th Message Understanding Conference (MUC-7) (1998)

[12] Farmakiotou, D., Karkaletsis, V., Koutsias, K., Sigletos, G., Spyropoulos, C.D., and Stamatopoulos, P.: *Rule-based Named Entity Recognition for Greek Financial Texts*. In Proceedings of the International Conference on Computational Lexicography and Multimedia Dictionaries COMLEX 2000 (2000)

[13] Grishman, R.: *Information Extraction: Techniques and Challenges*, Lecture Notes in Computer Science Vol. 1299, Springer-Verlag (1997)

[14] Iwanska, L., et al.: *Wayne State University:Description of the UNO natural language processing system as used for MUC-6*, In Proceedings of the 6th Message Understanding Conference (MUC-6) (1995)

[15] Mikheev, A., Grover, C., and Moen, M.: *Description of the LTG System Used for MUC-7,* In Proceedings of the 7th Message Understanding Conference (MUC-7) (1998)

[16] Morgan, R., et al.: *Description of the LOLITA system as used for MUC-6*, In Proceedings of the 6th Message Understanding Conference (MUC-6) (1995)

[17] Savitri, S.: *Analisa Struktur Kalimat Bahasa Indonesia dengan Menggunakan Pengurai Kalimat berbasis Linguistic String Analysis*, final project report, Fasilkom UI, Depok (1999) (in Indonesian)

[18] Sekine, S., Grishman, R., and Shinnou, H.: *A Decision Tree Method for Finding and Classifying Names in Japanese Texts*, Proceedings of the Sixth Workshop on Very Large Corpora, Montreal, Canada (1998)

[19] Tur, G., Hakkani-Tur, D.Z., and Oflazer, K.: *Name Tagging Using Lexical, Contextual, and Morphological Information*, Workshop on Information Extraction Meets Corpus Linguistics LREC-2000, 2nd International Conf. Language Resources and Evaluation, Athens, Greece (2000)

[20] Wahyudi, G.:Pengenalan Entitas Bernama berdasarkan Informasi Kontekstual, Morfologi dan Kelas Kata, final project report, Fasilkom UI, Depok (2004) (in Indonesian)

Bias Management of Bayesian Network Classifiers

Gladys Castillo[1,2] and João Gama[1,3]

[1] LIACC, University of Porto, Portugal
[2] Department of Mathematics, University of Aveiro, Portugal
[3] FEP, University of Porto, Portugal
gladys@mat.ua.pt, jgama@liacc.up.pt

Abstract. The purpose of this paper is to describe an adaptive algorithm for improving the performance of Bayesian Network Classifiers (BNCs) in an on-line learning framework. Instead of choosing a priori a particular model class of BNCs, our adaptive algorithm scales up the model's complexity by gradually increasing the number of allowable dependencies among features. Starting with the simple Naïve Bayes structure, it uses simple decision rules based on qualitative information about the performance's dynamics to decide when it makes sense to do the next move in the spectrum of feature dependencies and to start searching for a more complex classifier. Results in conducted experiments using the class of Dependence Bayesian Classifiers on three large datasets show that our algorithm is able to select a model with the appropriate complexity for the current amount of training data, thus balancing the computational cost of updating a model with the benefits of increasing in accuracy.

Keywords: Bias Management, Bayesian Classifiers, Machine Learning.

1 Introduction

Efficient learning algorithms usually involve an artful trade-off of *bias* vs. *variance*. If we choose a model that is too complex for the amount of training data we have, it will *overfit* the data. The model has too much variance. Otherwise, if the model is too simple, it cannot capture the true structure in the data, it will *underfit* the data. The model has too much bias. We can improve the performance of learning algorithms if we reduce either bias or variance. When we have few training data we can reduce variance by using simpler models while not increasing our bias too much. However, as it was shown in [2] as training set size increases variance will decrease and this will become a less significant part of the error. In this case, we must place more focus on bias management.

A well-studied and effective classifier is Naïve Bayes (NB). Although NB has a high bias due to its strong feature independence assumptions, its performance is compensated by its high variance management, thus producing accurate classification. **B**ayesian **N**etwork **C**lassifiers (BNCs) have been the natural choice

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 70–83, 2005.

for improving the predictive capability of NB. For instance, TAN classifiers [4] reduce the NB's bias by allowing the features to form a tree. In this paper, we examine an adaptive algorithm for improving the performance of BNCs in an on-line learning framework. Instead of choosing a priori a particular model class of BNCs, we propose to scale up the model's complexity by gradually increasing the number of allowable dependencies among features. If we scale up complexity slowly enough, the use of more training data will reduce bias at a rate that also reduces variance and consequently the classification error. This structure regularization leads to the selection of simpler models at earlier learning steps and of more complex structures as the learning process advances, thus avoiding the problems caused by either too much bias or too much variance. Starting with the simple NB, we use simple heuristics based on the performance's dynamics to decide about the next move in the spectrum of feature dependencies. This bias management attempts to select models with the appropriate complexity for the current amount of data, thus balancing the computational cost of updating a model with the benefits of increasing in accuracy.

We choose the class of *k*-**D**ependence **B**ayesian **C**lassifiers (*k*-DBC) for illustrating our approach. A *k*-DBC [11] is a Bayesian Network, which contains the structure of NB and allows each feature to have a maximum of *k* feature nodes as parents. This class is very suitable for our proposal. By varying *k* we can obtain classifiers that move smoothly along the spectrum of feature dependencies, thus providing a flexible control over the model's complexity. For instance, NB is a 0-DBC, TAN is a 1-DBC, etc. Although the adaptive algorithm is presented here for the family of *k*-DBC classifiers, we believe that its underlying principles can be easily adapted for learning other classifier's classes with flexible control over their complexity.

This paper is organized as follows: Section 2 briefly reviews the problem of learning Bayesian Network Classifiers and provides the learning algorithm for the class of *k*-DBCs. In Section 3 we describe our adaptive algorithm in an on-line learning framework. Next, in Section 4 we describe the experiments we conducted that demonstrate the effectiveness of our adaptive approach. Finally, in Section 5 we conclude with a summary of the paper.

2 Learning k-Dependence Bayesian Classifiers

Bayesian Networks (BNs) are probabilistic graphical models that provide a sound theoretical framework to represent and manipulate probabilistic dependencies in a domain. Formally, a BN over a set of random variables $\mathbf{X} = \{X_1, X_2, , X_n\}$ is a pair $BN = (S, \mathbf{\Theta_S})$ where the first component is a directed acyclic graph with a node for each variable and the second component is the set of parameters that quantifies the network. The arcs in the structure S represent direct dependencies between variables. Assuming discrete variables, each $\Theta_i = P(X_i|\mathbf{Pa_i}) \in \mathbf{\Theta_S}$ represents a conditional probability table that enumerates the probabilities for each possible value $x_k \in X_i$ and $pa_j \in \mathbf{Pa_i}$ where $\mathbf{Pa_i}$ represents the set of parents of X_i.

In classification problems the domain variables are partitioned into features $\mathbf{F} = \{X_1, X_2, , X_n\}$ and the class variable C. A NB classifier is a BN over $(\mathbf{F} \cup \{C\})$ with a simple structure that has the class node C as the parent node of all other feature nodes in $\mathbf{F}$. A k-DBC [11] is a BN which contains the structure of NB and allows each feature X_i to have a maximum of k feature nodes as parents. k-DBCs represents in a single class, a full spectrum of allowable dependence in a given probabilistic model with the NB classifier at the most restrictive end and the full Bayesian Network at the most general extreme.

Suppose we have a set $\mathcal{M}$ of BNC model classes (e.g. NB, TAN, unrestricted BNs, etc.) and a training dataset $\mathcal{D}$ of labelled i.i.d. examples of (F, C). Since the quality of a BNC is defined in terms of its *predictive accuracy*, given the data $\mathcal{D}$ and the set $\mathcal{M}$ of BNC's hypotheses, the problem of learning BNCs is to find a BNC that provides the best classifications for future data. This learning problem - a *model selection problem* - can be approached as a discrete optimization problem where a score that measures the quality of each hypothesis is optimized in the space of feasible hypotheses. A procedure to solve this discrete optimization problem is essentially a search algorithm that explores the space of candidate hypotheses while optimizing the score. In most cases, the search space is large and an exhaustive search is impractical. One way to handle this problem is to develop heuristic search algorithms that search for a solution which is reasonably close to optimal.

There are three main factors that affect the performance of score-based approaches to model selection: the *score* and the *search method* used to learn the structure and the *estimator* used to learn the parameters. Next we describe these three factors in learning k-DBCs.

2.1 Search Algorithm

Instead of using the learning algorithm proposed by Sahami [11] based on the computation of the conditional mutual information, we apply, in conjunction with a score, a Hill Climbing procedure. Hill Climbing improves the score by *adding/removing/reversing* arcs among attributes, subject to never introducing a cycle. This process continues until no more operations improve the score.

2.2 Scores

When BNs are used for classification, we are interested in the resulting predictive distribution yielding accurate predictions for future data. We compare three frequently used scores in learning BNs: BDeu, MDL and AIC; to the prequential score (Preq) as described in [7]. BDeu, MDL and AIC are optimized for a particular loss function based on the joint distribution while Preq is optimized for classification. AIC and MDL are both derived from information theory and prefer simpler models. Minimizing AIC is approximately equivalent to minimizing the expected K-L divergence between the true model and the fitted model. MDL principle attempts to describe the data using a minimum encoding approach. BDeu is a Bayesian score, the marginal likelihood with uniform priors as proposed in

[3]. Preq is computed *predictively* and *sequentially* through a sequential updating of the predictive distribution. Alternative structures are compared by measuring their cumulative loss. While it is known that standard scores perform worse in classification than scores based on the classification error (e.g. see [7]), we are more interested in investigating how different scores handle the *bias-variance*, *complexity-performance* trade-offs in incremental learning of k-DBCs.

2.3 Parameter Estimation

We use the Bayesian estimates for parameters as described in [1]. In addition, we optionally use an extended version of Iterative Bayes (IB) [5] for parameter refinement. IB iteratively cycles through all the given examples. For each example, the corresponding entries in the contingency tables are updated so as to increase the confidence on the correct class. The procedure is given an initial error for the given examples. The iterative process stops when one of the following cases occurs: *i*) the current error is greater than the previous error; *ii*) the error remains stationary for three consecutive times; *iii*) a maximum number of allowed iterations is reached. In any of the cases, the model returned is that which attains the best results during the whole iterative process.

3 The Adaptive Algorithm for Learning k-DBCs

In this section we describe our adaptive algorithm for learning k-DBCs in an online framework. We provide our algorithm with a dataset of labelled examples and the kMax value for the maximum allowable number of feature dependencies. We assume the environment is stationary, data arrives in batches and a unique k-DBC model is maintained. The pseudo-code for the algorithm is presented in Figure 1. At each learning step, the learner accepts a batch of examples and classifies them using the current model. Next, the current performance is assessed and the model is then adapted according to the estimated performance' state.

An efficient adaptive algorithm for supervised learning must be able, above all, to improve its predictive accuracy over time, while minimizing the cost of updating. BNs suffer from several drawbacks for updating purposes. While sequential updating of the parameters is straightforward (if data is complete); updating the structure is a more costly task. In previous work different approaches have been carried out in incremental learning of BNs by optimizing the learning algorithms and/or the memory space (see [10] for a survey). The basic idea of our approach is that we can improve the performance while reducing the cost of updating if: *i*) in each learning step we choose a model with the appropriate complexity for the amount of training data we have; *ii*) we try to use new data to primarily adapt the parameters and only when it is really necessary to adapt the structure. As a result, our strategy leads to the selection of simpler models at earlier learning steps and gradually increases the model's complexity as more and more data becomes available. This bias control attempts to avoid *overfitting*

```
procedure AdaptiveOnlinekDBCs(data,kMax)
init: model   InitAsNaiveBayes()
for each new incoming batch of examples
    predictions <- predict(model,batch)
    observed <- getFeedback(batch)
    performanceState <- assesPerformance(predictions, observed)
    if (performanceState != IS_SATISFACTORY)then
        adapt(model, examples, FIRST_LEVEL)
    if (performanceState == STOP_IMPROVING) then
        adapt(model, examples, SECOND_LEVEL)
        if (Not change(model.structure)) then
           adapt(model, examples, THIRD_LEVEL)
end for
end procedure
```

Fig. 1. Pseudo-code for the adaptive on-line algorithm for k-DBCs

or *underfitting* of the current model to the actual data. Next, we describe the two main aspects of our adaptive algorithm: the *adaptation policy* and the *control policy*.

3.1 Adaptation Policy

The *adaptation policy* is characterized by a gradual adaptation of the model using three levels so that increasing the adaptation level increases the cost of updating:

- FIRST LEVEL: only the parameters are updated with new data
- SECOND LEVEL: the current structure is adapted by searching for new dependencies among attributes
- THIRD LEVEL: if it is still possible, the maximum number of allowable dependencies is increased by one, and the current structure is once again adapted.

The pseudo-code for the adaptation procedure is presented in Figure 2. In the absence of any information about the true model underlying the data, we initialize the classifier to the simple NB ($k = 0$). Whenever we obtain new data, we first try to improve NB by adapting only its parameters. Only when we obtain some evidences indicating that the performance of the NB stops improving in the desirable tempo, we move to a more costly level of adaptation: adapting the structure. We increment k by one and start searching a 1-DBC by finding 1-dependence among attributes. At this time point, we must have more data available which allows the search procedure to find new dependencies. Next, the algorithm continues to perform only parameter adaptation, until there will be again evidences that the performance of the current classifier

```
procedure Adapt(model, examples, level)
   switch level:
     case FIRST_LEVEL:
       performAdaptation(model, examples, UPDATE_PARAMETERS)
       if (bUseIterativeBayes) then
         performAdaptation(model, examples, REFINE_PARAMETERS)
     case SECOND_LEVEL:
         performAdaptation(model, examples, ADAPT_STRUCTURE)
     case THIRD_LEVEL:
       if (augmentDepIsPossible(model))then
         augmentMaxNrAllowableDependencies(model)
         performAdaptation(model, examples, ADAPT_STRUCTURE)
   end switch
 return model
end procedure
```

Fig. 2. Pseudo-code for the adaptive algorithm

stops improving. In this case, we try to adapt the current structure. Only if the resulting structure remains the same, we move to the third level of adaptation by incrementing the maximum number of allowable dependencies, k, by one (if this is still possible, i.e. if $k < \mathsf{kMax}$) and searching for new dependencies. This process continues until the performance reaches the desirable level.

3.2 Control Policy

The *control policy* defines the criteria for tracking two situations: i) at what time point do we move from the first level of adaptation to the second level, i.e., when do we start adapting the structure? ii) at what time point do we stop doing any adaptation? If we detect that the performance of the current model no longer improves in a desirable tempo then we start adapting the structure. On the other hand, if we detect that the performance has already reached the desirable level, we stop adapting the model.

Assume that feedback can be obtained and that for each batch, we can evaluate the error rate e_{batch}, the proportion of misclassified examples in a batch. We monitor e_{batch} obtained for different batches as an indicator of the performance at different points in time. As stated, we initialize the structure to NB. Because of its simplicity, NB learns very quickly, which is reflected in the behavior of the batch error. At earlier learning steps, it exhibits a shorter downward trend with a steeper slope of descent. However, as time increases, the steepness of the slope will decrease, approaching zero. We use the Sen's slope estimator [12] for assessing the trend strength. At each t^{th} learning step, we use only the most recent p batch errors for dynamically assessing the decreasing slope (we set p to 5). To estimate the Sen's slope we compute the slopes of each pair of observed errors $e_{batch}[t_i], e_{batch}[t_j]$ for $(t_i > t_j)$ where the slope is defined as $(e_{batch}[t_i] - e_{batch}[t_j])/(t_i - t_j)$. The Sen's slope is then the median value of the resulting slopes. The rule is then straightforward. If the slope is sufficiently close to zero, then we assume that the performance of the current model no longer improves:

$$IF \;\; SenSlope(e_{batch}[t-p+1:t]) \leq slope_{threshold}$$

$$THEN \;\; performanceState \leftarrow StopImproving$$

At subsequent learning steps it results more difficult to apply this kind of trend analysis using successive error values. Notice that as time increases, batch error values fluctuate around a certain level, decreasing slowly with a slope approaching zero. Instead, we proceed in the following way: first, the parameters are updated using new examples. Then, we once again assess e_{batch} using the adapted model. Assume that $e'_{batch}[t]$ and $e''_{batch}[t]$ are the batch errors obtained before and after adaptation, respectively. Whenever we obtain a decrease of the batch error after adaptation, it would be a straightforward idea to consider that the learner is still able to learn about the current target concept using the current model's structure. Otherwise, if for a pre-defined number of consecutive times after adaptation the error does not improve then we assume that the performance no longer improves using the current structure:

$$IF \;\; consecCounter(e''_{batch}[t] \geq e'_{batch}[t]) = maxTimes$$

$$THEN \;\; performanceState \leftarrow StopImproving$$

Further model adaptations will continue until the performance reaches the desirable level. Given a threshold level for the batch error, we assume that the performance is satisfactory if for a fixed number of consecutive times $e_{batch} \leq error_{threshold}$.

4 Empirical Study

Primarily, we want to investigate if our adaptive algorithm is able to scale up the model's complexity of k-DBCs while improving its performance over time. With this aim, we carried out an empirical study for evaluating the performance of k-DBCs and NB induced incrementally from scratch against our adaptive approach for four scores on three large datasets.

4.1 Experimental Setup

We used two underlying learning algorithms to induce k-DBCs: NB ($k = 0$) and hill-climbing ($k > 0$) with BDeu, MDL, AIC and Preq as described in section 2. We used only arc additions and deletions. All the learning algorithms were implemented using Weka's classes for BNCs ([1],[13]). Since we use different scores for the same learning algorithm, this helps in ensuring that any differences in performance are due to the differences in the scores, and not to differences in the underlying algorithm.

We evaluated the learning algorithms on three datasets: *balance*, *nursery* and *adult*. Since we needed datasets with large number of examples to better explore the behaviour of incremental algorithms, we randomly generated artificial large samples of 10000 examples for balance using its well-known underlying rules. We used the nursery dataset from the UCI repository and a discretized version of the adult dataset available on-line at *http://www.cs.helsinki.fi/u/p̃kontkan/Data/*. We removed instances with missing values from the datasets. Thus, we used 12800 instances for nursery (128 learning steps) and 16000 instances for adult (160 learning steps), respectively.

We evaluated two versions of the AdaptiveOnlinekDBCs algorithm. Unlike Adap1, Adap2 additionally implements IB (section 2.3). We set kMax=3 for balance, kMax=5 for *nursery* and adult, $slope_{threshold} = error_{threshold} = 0$ and $maxTimes = 3$. To serve as baselines of our adaptive algorithms, we evaluated the performance of NB and k-DBCs (varying k), inducing them incrementally from scratch: at the t^{th} learning step we used the first t batches as training data and the examples of the next $(t+1)^{th}$ batch as test data. We use batches of 100 examples. At each learning step, the performance was measured as the average of the accuracy over 10 runs.

4.2 Cost of Updating vs. Performance

Table 4 compares the relative significant gains of the predictive accuracy averaged over 10 runs of k-DBCs and Adap1-2 in conjunction with the four scores with respect to NB at different learning steps. Figure 3 compares the performance over time of all the algorithms for the three datasets. Table 1 shows the number of adaptations performed in the structure per data set, score and adaptive algorithm.

In most cases results show that adaptive algorithms perform at least as well as the best k-DBC at each learning step. In general, Adap1-2 significantly improve the performance of NB over time while reducing the cost of updating, as it is shown by the small number of adaptations performed in the structure during the whole learning process. However, the best results were obtained with the *nursery* and the *adult* datasets. Note that for *balance*, as time increases, the best model for all the scores, except for MDL, is a 3-DBC. Since the balance domain is easier to learn, adaptive algorithms can get trapped in less complex structures than the optimal one while progressing to improve their performances. Unlike balance, for nursery and adult the best results are obtained with Adap2. Moreover, for all the scores and datasets the number of adaptations performed in the structures using Adap2 is considerably less than using Adap1 (see Table 1). As it was shown in [5], the reduction of the error rate observed with IB is mainly due to a reduction on the bias component, which explains the obtained results. This means that Adap2 ensures the best balance between the cost of updating and the gain in performance.

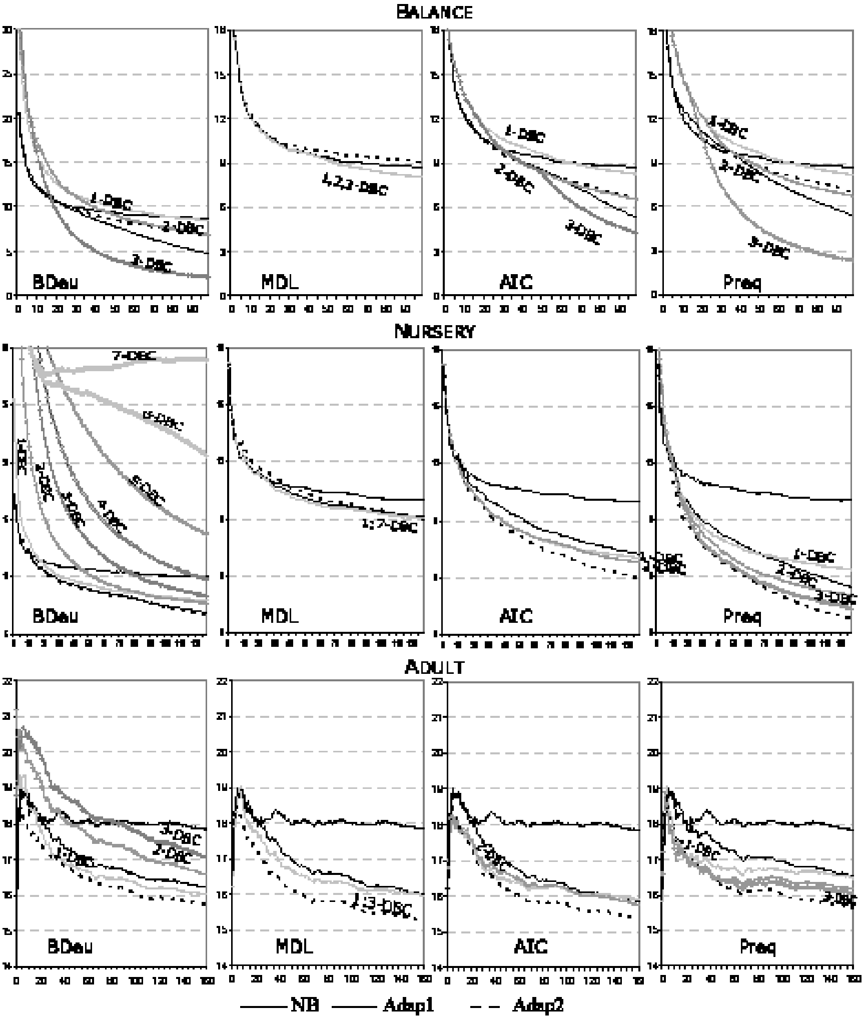


Fig. 3. Error rate for NB, k-DBCs and Adap1-2 for the four scores over time. At the $t^t h$ learning step, # training examples = 100 $*$ t; # test examples = 100.

4.3 Model Complexity vs. Performance

There is a bias-variance trade-off in choosing the appropriate complexity of the model. We used the bias-variance decomposition of the error as proposed in [6] to investigate how different scores handle the bias-variance trade-off in incremental learning of k-DBCs. Due to space limitations we only show the results on the *nursery* dataset at two point times: t = 10 and t = 120 in figure 4. Plots on the left show the training and test errors as a function of k-DBC models. Each point in a line represents the training or test error of the related k-DBC for

Table 1. Number of adaptive actions per data set, score and adaptive algorithm

		balance		nursery		adult	
Score	Algor.	Ref.Str.	Aug.Dep	Ref.Str.	Aug.Dep	Ref.Str.	Aug.Dep
BDeu	Adap1	3.0±0.0	3.0±0.0	3.2±0.5	3.2±0.5	1.8±0.5	1.8±0.5
	Adap2	2.1±0.9	2.1±0.9	3.4±0.6	3.4±0.6	1.2±0.5	1.2±0.5
MDL	Adap1	16.7±0.0	3.0±0.0	23.8±2.8	5.0±0.0	14.0±2.0	4.0±0.0
	Adap2	1.2±0.6	1.1±0.3	15.6±3.8	5.0±0.0	3.8±0.8	2.4±1.6
AIC	Adap1	3.7±0.7	3.0±0.0	14.4±2.0	5.0±0.0	4.6±1.1	3.4±0.9
	Adap2	2.2±0.9	2.2±0.9	13.2±2.1	5.0±0.0	2.4±0.6	1.6±0.6
Preq	Adap1	3.5±0.5	3.0±0.0	5.0±1.9	3.6±1.1	3.4±0.6	2.8±0.5
	Adap2	2.4±1.2	2.2±0.9	5.2±1.9	3.8±1.1	1.6±0.6	1.4±0.6

a particular score. The first points in the lines represent the errors of NB. In each learning step, given a particular score there is an optimal model class that gives minimum test error. For Preq and BDeu, the optimal models are 1-DBC at $t = 10$ and 3-DBC at $t = 120$, respectively. For MDL and AIC, all k-DBCs present identical results starting from some k. Results in Table 3 suggest that found models are all identical.

Pictures on the right show the *bias-variance* decomposition of the test error for all scores. Results show that varying k, the score and the training set size can have different effects on bias and variance. The best results were obtained with Preq due to a more optimal bias management. On the other hand, BDeu consistently favors the dependent structure over the independent one, thus finding maximal models (see Table 3). As you can see, if the class model is more

Table 2. The values of k averaged over 10 runs at $t = 10$ and $t = 120$ for the nursery dataset

	Adap1				Adap2			
t	BDeu	MDL	AIC	Preq	BDeu	MDL	AIC	Preq
10	0.8	0.8	0.8	0.8	1	1	1	1
120	3.2	5	5	3.6	3.2	5	5	3.4

Table 3. The number of added arcs to the NB structure averaged over 10 runs for the nursery dataset at two time points. K1-7 represent 1-7-DBCs, A1-2 - Adap1, Adap2, respectively

	1000 training examples									12000 training examples								
Score	K1	K2	K3	K4	K5	K6	K7	A1	A2	K1	K2	K3	K4	K5	K6	K7	A1	A2
BDeu	7	13	18	22	25	27	28	4.2	5.6	7	13	18	22	25	27	28	18.8	18.6
MDL	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2
AIC	1.4	1.4	1.4	1.4	1.4	1.4	1.4	0.8	1.2	7	9.4	9.4	9.4	9.4	9.4	9.4	10.4	9.8
Preq	6.4	8	8	8	8	8	8	3.4	4.2	7	12.6	17.4	18	18.4	18.4	18.4	17.6	17.8

Table 4. Relative significative gains of the predictive accuracy of k-DBCs, Adap1-2 with respect to NB. (+) indicates significative gain using a paired t-test at the 5% level , *no* indicates there is no significative gain.

Balance		10	20	30	40	50	60	70	80	90	100
	NB	87.52	88.62	89.04	89.22	89.34	89.52	89.6	89.66	89.81	89.87
BDeu	1-DBC	no	no	no	no	no	no	no	no	no	(+)0.29
	2-DBC	no	no	no	no	no	(+)0.66	(+)1.06	(+)1.35	(+)1.63	(+)1.86
	3-DBC	no	(+)0.91	(+)3.14	(+)4.5	(+)5.31	(+)5.67	(+)6.06	(+)6.29	(+)6.52	(+)6.65
	Adap1	no	no	no	(+)0.99	(+)1.57	(+)2.03	(+)2.61	(+)3.12	(+)3.6	(+)3.95
	Adap2	no	no	no	(+)0.53	(+)0.86	(+)1.03	(+)1.25	(+)1.52	(+)1.82	(+)2.11
MDL	1-DBC	no	no	no	no	(+)0.19	(+)0.3	(+)0.44	(+)0.55	(+)0.66	(+)0.72
	2-DBC	no	no	no	no	(+)0.19	(+)0.3	(+)0.44	(+)0.55	(+)0.66	(+)0.72
	3-DBC	no	no	no	no	(+)0.19	(+)0.3	(+)0.44	(+)0.55	(+)0.66	(+)0.72
	Adap1	no	no	no	no	no	(+)0.28	(+)0.43	(+)0.55	(+)0.65	(+)0.71
	Adap2	no	no	no	no	no	no	no	no	no	no
AIC	1-DBC	no	no	no	no	no	no	no	(+)0.28	(+)0.42	(+)0.5
	2-DBC	no	no	no	(+)0.55	(+)0.98	(+)1.26	(+)1.58	(+)1.81	(+)2.01	(+)2.2
	3-DBC	no	no	no	(+)0.55	(+)1.04	(+)2.08	(+)2.98	(+)3.6	(+)4.13	(+)4.48
	Adap1	no	no	no	(+)0.45	(+)0.83	(+)1.14	(+)1.74	(+)2.32	(+)2.95	(+)3.4
	Adap2	no	no	no	(+)0.6	(+)0.9	(+)1.12	(+)1.36	(+)1.64	(+)1.93	(+)2.19
Preq	1-DBC	no	no	no	no	no	no	no	(+)0.27	(+)0.46	(+)0.57
	2-DBC	no	no	no	no	(+)0.65	(+)1.01	(+)1.33	(+)1.6	(+)1.84	(+)2.03
	3-DBC	no	no	(+)2.25	(+)3.82	(+)4.77	(+)5.22	(+)5.67	(+)5.95	(+)6.22	(+)6.38
	Adap1	no	no	no	no	(+)1.01	(+)1.47	(+)2.05	(+)2.49	(+)2.93	(+)3.34
	Adap2	no	no	no	no	(+)0.63	(+)0.82	(+)1.04	(+)1.17	(+)1.45	(+)1.77

Nursery		10	20	30	40	50	60	80	100	110	128
	NB	87.52	88.62	89.04	89.22	89.34	89.52	89.66	89.97	89.98	89.96
BDeu	1-DBC	no	no	(+)0.93	(+)1.12	(+)1.36	(+)1.51	(+)1.75	(+)1.90	(+)1.98	(+)2.18
	2-DBC	no	no	no	no	(+)0.45	(+)0.97	(+)1.64	(+)2.00	(+)2.12	(+)2.35
	3-DBC	no	no	no	no	no	no	no	(+)0.95	(+)1.23	(+)1.73
	Adap1	no	(+)0.80	(+)1.19	(+)1.46	(+)1.70	(+)1.84	(+)2.25	(+)2.65	(+)2.83	(+)3.16
	Adap2	no	(+)0.91	(+)1.38	(+)1.62	(+)1.86	(+)1.97	(+)2.26	(+)2.77	(+)3.02	(+)3.39
MDL	1-DBC	no	no	(+)0.32	(+)0.51	(+)0.71	(+)0.77	(+)0.82	(+)0.84	(+)0.88	(+)1.02
	2-DBC	no	no	(+)0.32	(+)0.51	(+)0.71	(+)0.77	(+)0.82	(+)0.84	(+)0.88	(+)1.02
	3-DBC	no	no	(+)0.32	(+)0.51	(+)0.71	(+)0.77	(+)0.82	(+)0.84	(+)0.88	(+)1.02
	Adap1	no	no	no	no	(+)0.51	(+)0.6	(+)0.68	(+)0.73	(+)0.77	(+)0.92
	Adap2	no	no	no	no	(+)0.28	(+)0.42	(+)0.56	(+)0.66	(+)0.77	(+)1.05
AIC	1-DBC	(+)0.54	(+)0.89	(+)1.46	(+)1.87	(+)2.19	(+)2.38	(+)2.70	(+)2.80	(+)2.86	(+)3.02
	2-DBC	(+)0.54	(+)0.89	(+)1.46	(+)1.87	(+)2.19	(+)2.39	(+)2.66	(+)2.95	(+)3.03	(+)3.23
	3-DBC	(+)0.54	(+)0.89	(+)1.46	(+)1.87	(+)2.19	(+)2.39	(+)2.66	(+)2.95	(+)3.03	(+)3.23
	Adap1	no	(+)0.58	(+)0.91	(+)1.12	(+)1.47	(+)1.78	(+)2.15	(+)2.39	(+)2.50	(+)2.78
	Adap2	no	(+)1.09	(+)1.61	(+)2.10	(+)2.44	(+)2.78	(+)3.22	(+)3.62	(+)3.76	(+)4.10
Preq	1-DBC	no	(+)1.59	(+)2.31	(+)2.72	(+)2.96	(+)3.13	(+)3.35	(+)3.46	(+)3.53	(+)3.72
	2-DBC	no	(+)1.86	(+)2.69	(+)3.26	(+)3.75	(+)4.06	(+)4.50	(+)4.74	(+)4.82	(+)5.05
	2-DBC	no	(+)2.06	(+)3.14	(+)3.64	(+)4.10	(+)4.42	(+)4.99	(+)5.28	(+)5.41	(+)5.73
	Adap1	(+)0.34	(+)1.38	(+)1.98	(+)2.37	(+)2.70	(+)2.96	(+)3.49	(+)3.94	(+)4.15	(+)4.62
	Adap2	no	(+)2.56	(+)3.33	(+)3.80	(+)4.24	(+)4.58	(+)5.29	(+)5.76	(+)5.93	(+)6.29

Adult		10	20	40	60	80	90	100	120	140	168
	NB	81.14	81.91	81.69	81.89	82.00	81.93	81.99	82.01	82.04	82.16
BDeu	1-DBC	no	(+)0.15	(+)1.34	(+)1.46	(+)1.54	(+)1.62	(+)1.65	(+)1.80	(+)1.86	(+)1.82
	2-DBC	no	no	no	no	(+)0.59	(+)0.70	(+)0.80	(+)1.08	(+)1.17	(+)1.26
	3-DBC	no	no	no	no	no	no	no	(+)0.48	(+)0.63	(+)0.75
	Adap1	no	no	(+)0.91	(+)1.08	(+)1.24	(+)1.35	(+)1.41	(+)1.56	(+)1.61	(+)1.62
	Adap2	(+)0.98	(+)0.49	(+)1.43	(+)1.62	(+)1.77	(+)1.86	(+)1.93	(+)2.09	(+)2.11	(+)2.09
MDL	1-DBC	no	(+)0.34	(+)1.36	(+)1.58	(+)1.66	(+)1.72	(+)1.74	(+)1.86	(+)1.91	(+)1.87
	2-DBC	no	(+)0.34	(+)1.36	(+)1.59	(+)1.66	(+)1.72	(+)1.75	(+)1.88	(+)1.90	(+)1.87
	3-DBC	no	(+)0.34	(+)1.36	(+)1.59	(+)1.66	(+)1.72	(+)1.75	(+)1.88	(+)1.90	(+)1.87
	Adap1	no	no	(+)0.95	(+)1.31	(+)1.40	(+)1.51	(+)1.55	(+)1.75	(+)1.81	(+)1.83
	Adap2	(+)1.00	(+)0.78	(+)1.86	(+)2.14	(+)2.17	(+)2.27	(+)2.32	(+)2.52	(+)2.57	(+)2.57
AIC	1-DBC	(+)1.06	(+)0.60	(+)1.71	(+)1.77	(+)1.78	(+)1.79	(+)1.84	(+)1.96	(+)1.98	(+)1.96
	2-DBC	(+)1.04	(+)0.55	(+)1.51	(+)1.65	(+)1.73	(+)1.77	(+)1.82	(+)1.99	(+)2.07	(+)2.10
	3-DBC	(+)1.04	(+)0.55	(+)1.51	(+)1.65	(+)1.73	(+)1.77	(+)1.82	(+)1.99	(+)2.07	(+)2.10
	Adap1	no	no	(+)1.08	(+)1.39	(+)1.56	(+)1.66	(+)1.70	(+)1.89	(+)1.97	(+)2.00
	Adap2	(+)0.94	(+)0.61	(+)1.86	(+)2.11	(+)2.17	(+)2.24	(+)2.25	(+)2.42	(+)2.49	(+)2.48
Preq	1-DBC	no	(+)0.92	(+)1.41	(+)1.35	(+)1.28	(+)1.33	(+)1.31	(+)1.35	(+)1.37	(+)1.37
	2-DBC	(+)1.44	(+)1.06	(+)1.67	(+)1.66	(+)1.54	(+)1.58	(+)1.54	(+)1.67	(+)1.69	(+)1.68
	3-DBC	(+)1.52	(+)1.00	(+)1.71	(+)1.77	(+)1.67	(+)1.72	(+)1.68	(+)1.79	(+)1.81	(+)1.78
	Adap1	no	no	(+)0.64	(+)0.85	(+)0.93	(+)1.01	(+)1.03	(+)1.19	(+)1.26	(+)1.30
	Adap2	(+)1.00	(+)0.93	(+)1.70	(+)1.91	(+)1.88	(+)1.93	(+)1.97	(+)2.14	(+)2.19	(+)2.22

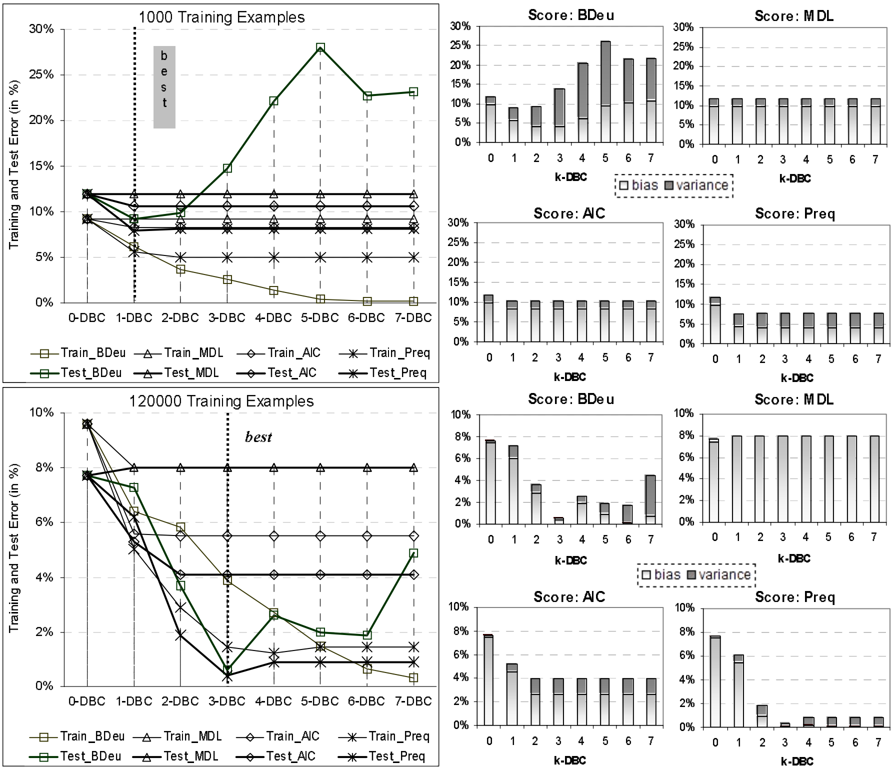


Fig. 4. Training-test errors and bias-variance decompositions of k-DBCs varying k at two time points

complex than the optimal model, BDeu leads to severe overfitting due to increase in variance. However, as training set size increases, the variance decreases for all k-DBCs, thus reducing the test error and the overfitting problem. In contrast, MDL and AIC find models simpler than the optimal model, thus underfitting the data. Both scores increase in bias, especially, if the class model is less complex than the optimal model. Because MDL penalizes complexity more severely that AIC does, the model with the least MDL will tend to be simpler than the model with the most AIC. Notice that as training size increases AIC reduces the bias slightly. On the contrary, MDL increases it. As a result AIC outperforms MDL.

To provide evidences toward the hypothesis that our adaptive algorithm attempts to select the appropriate complexity of the model (i.e. the optimal model class) for the current amount of training data, we can look at table 2. At $t = 10$, for all scores, adaptive algorithms find a model with k approaching 1, i.e., a 1-DBC. At $t = 120$, for BDeu and Preq, they find a model with k approaching 3 (a 3-DBC). For MDL and AIC, they find a model with k =kMax, i.e., a 5-DBC. These results are consistent with the optimal model classes that we have found previously. Note that optimal k-DBCs present the lowest biases. Finally, results in Table 1 evidence that the number of adaptations performed in the structure for BDeu was always minimal when compared with other scores. On the contrary, the number of adaptations performed in the structures using MDL was always maximal. These results reflect the efforts made by our adaptive algorithms to control the overfitting and underfitting problems.

5 Conclusions

We have examined a practical adaptive learning algorithm for improving the performance of BNCs over time. The main idea is to scale up the model's complexity as training data increases by gradually increasing the number of allowable dependencies among features. This allows reducing both bias and variance and consequently the classification error. Starting with the simple NB, we use simple decision rules based on the performance dynamics to decide on the next move in the spectrum of feature dependencies and search for a more complex model. Therefore, as training set size increases, bias will decrease because we choose a more complex model and variance will also decrease because we use more examples to learn. Results in conducted experiments using the class of k-DBCs and a hill climbing learning algorithm in conjunction with four scores on three large datasets show that our adaptive algorithm in combination with IB performs an artful bias management for choosing the appropriate complexity of the model.

Our adaptation policy is characterized by a gradual adaptation of the model using three levels so that increasing the adaptation level increases the cost of updating. We attempt to use new data to primarily adapt the parameters and only if this is really necessary, to adapt the structure. Since updating the structure is a costly task, this way we reduce the computational cost of updating while improving the performance. Results in conducted experiments show that adaptive algorithms significantly improve the performance of NB over time and that they perform no worse than the best k-DBC while reducing the cost of updating as it is shown by the small number of adaptations performed on the structure during the whole learning process in contrast to the great number of adaptations performed on the structure of k-DBCs when they were induced incrementally from scratch.

Although we have used only three datasets for evaluation, the results obtained here encourage us to continue this work thus to be able to improve the adaptation and control policies involved in our adaptive algorithm. One of the crucial question that we will focus in the future will be to investigate several criteria for determining when we should stop the learning process according to the observed performance, instead of using a given threshold level for the batch error. Future work will also involve additional experimentation with more large datasets in order to obtain more evidences on the effectiveness of our adaptive system.

Finally, although the adaptive algorithm is presented here for the family of k-DBC classifiers, we believe that its underlying principles can be easily adapted for learning other classifier's model classes (e.g. decision trees [8], neural networks using different topologies [9]) with a *hierarchical* and *increasing* control over their complexity.

Acknowledgments

Thanks to the financial support given by the FEDER, the Plurianual support attributed to LIACC, and project ALES II (POSI/EIA/55340/2004).

References

1. Bouckaert, R.: Bayesian Network Classifiers in Weka (2004), Technical Report 14/2004. Computer Science Department. University of Waikato. (2004)
2. Brian, D., Webb, G.: The need for Low Bias Algorithms in Classification Learning from Large Data Sets, In Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2002), Springer-Verlag (2002) 62: 73
3. Buntine, W.: Theory Refinement on Bayesian Networks. In Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence, (1991) **52**: 60 4.
4. Friedman, N., Geiger, D. and Goldszmidt, M.: Bayesian Network Classifiers. Machine Learning **4** (1997) 131:161 5.
5. Gama, J.: Iterative Bayes. In Intelligent Data Analysis, **4** (2000) 475:488, IOS Press 6.
6. Kohavi, R., Wolpert, D.: Bias Plus Variance Decomposition for Zero-One Loss Functions. In Proceedings of the 13th International Conference on Machine Learning (ICML'96), Morgan Kaufmann Publishers, (1999) 7.
7. Kontkanen, P., Myllymaki, P., Silander, T., Tirr, H: On Supervised Selection of Bayesian Networks. In Proceedings of the Fifteenth International Conference on Uncertainty in Artificial Intelligence (UAI'99), Morgan Kaufmann Publishers, (1999) 334:342
8. Quinlan, R.: *C4.5 Programs for Machine Learning.* Morgan Kaufmann Publishers, (1993).
9. Ripley, B.: *Pattern Recognition and Neural Networks.* Cambridge University Press, (1996).
10. Roure, J., Sangüesa,R.: Incremental Methods for Bayesian Network Learning. Research Report LSI-99-42-R. Software Department. Technical University of Catalonia, (1999).
11. Sahami, M.: Learning Limited Dependence Bayesian Classifiers, In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, Portland, OR, (1996) 335:338 10.
12. Sen, P.K.: Estimates of the regression coefficient based on Kendall's tau. In Journal of the American Statistical Association. **63** (1968) 1379:1389
13. Witten,I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, (2005).

A Bare Bones Approach to Literature-Based Discovery: An Analysis of the Raynaud's/Fish-Oil and Migraine-Magnesium Discoveries in Semantic Space

R. J. Cole[1] and P. D. Bruza[2]

[1] School of Info. Tech. and Elec. Eng.,
University of Queensland
rcole@itee.uq.edu.au
[2] Distributed Systems Technology Centre,
University of Queensland
bruza@dstc.edu.au

Abstract. Literature discovery can be characterized as a goal directed search for previously unknown implicit knowledge captured within a collection of scientific articles. Swanson's serendipitous discovery of a treatment for Raynaud's disease by dietary fish-oil while browsing Medline, an online collection of biomedical literature, exemplifies such a discovery. By means of a series of experiments, the impact of stop words, various weighting schemes, discovery mechanisms, and contextual reduction are studied in relation to replicating the Raynaud/fish-oil and migraine-magnesium discoveries by operational means. Two aspects of discovery were brought under focus: (i) the discovery of intermediate, or $B-$terms, and (ii) the discovery of indirect $A-C$ connections via the $B-$terms. A semantic space representation of the underlying corpus is computed and discoveries automated by computing associations between words in both higher and contextually reduced spaces. It was found that the discovery of $B-$terms and $A-C$ connections can be achieved to an encouraging degree with a standard stop word list. In addition, no single weighting scheme seems to suffice. Log-likelihood appears to be potentially effective for leading to the discovery of $B-$terms, whereas both odds ratio and simple co-occurrence frequencies both facilitate the discovery of $A-C$ connections. With regard to discovery mechanism, both semantic similarity (via cosine) and information flow computation seem promising for computing $A-C$ connections, but more research is needed to understand their relative strengths and weaknesses. Discovery in a contextually reduced semantic space revealed mixed results.

1 Introduction

In the mid-nineteen eighties, Don Swanson, made a chance discovery relating two discrete islands of literature, one related to the circulatory disease Raynaud's and the other with fish-oil. At the time Raynaud's did not have a cure or

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 84–98, 2005.

a general treatment. He formulated the hypothesis that dietary fish-oil might be beneficial to Raynaud's even though this information was not explicitly stated in either of the literatures surrounding Raynaud's or dietary fish-oils. His hypothesis was later corroborated by clinical studies [24]. Swanson also made subsequent discoveries, for example, the connection between migraine and magnesium.

The basic architecture of the discovery is denoted by $A - B - C$, where C denotes the phenomenon, e.g., Raynaud's, and A represents the potential cure, or treatment, e.g., fish oil [25]. The discovery between C and A is made by means of intermediate B-terms, e.g., "platelet aggregation", "vascular reactivity" and "blood viscosity". It is important to note that the connection between C and A is indirect. This article is about replicating the Raynaud/fish-oil discovery by operational means with the view of studying parameters such as stop words, weighting schemes and the like in order to see how they impact the effectiveness of discovery.

In terms of the $A - B - C$ architectures, two models of discovery have been identified. The open mode of discovery involves the generation of a hypothesis, for example, that "fish-oil may be a potential treatment for Raynaud's". In this article, the open mode of discovery is further refined. Firstly, there is the problem of identifying salient B-terms. Secondly, once salient B-terms have been identified, these are then used to make connections to potential $C-$terms. The closed mode of discovery involves the justification of the hypothesis. This article will focus on the open mode of discovery.

2 Related work

The problem of literature based discovery is exemplified by Swanson's Raynaud/ fish-oil discovery. This discovery highlighted the possible existence of several such hidden links in the literature. Swanson called the existence of such knowledge, *undiscovered public knowledge*. In a series of publications thereafter Swanson addressed the possibility of hidden hypotheses[20,22,23,21,24].

Swanson's attempt to automate the $A - B - C$ discovery resulted in the ARROWSMITH system [24]. This is a semi-automatic system and works as follows. First identify a concept C of interest such as the disease Raynaud's. Then finding a treatment for Raynaud's disease would involve inspecting all the concepts that are discussed in the same documents as those that discuss Raynaud's. These concepts are the B terms. Potential B-terms are identified by manual exclusion of terms deemed to be irrelevant to the phenomenon at hand and ranking the remainder by statistical means. Supporting this process is a sophisticated, manually updated stop word list. Then, using these B-terms the system finds A concepts that are discussed in association with the B-terms. The main disadvantage of ARROWSMITH is the amount of manual intervention required by the user in selecting the terms that need to be discarded from the B list and the A list.

Gordon & Lindsay [8,6] used lexical statistics, primarily TF*IDF. The authors used stemming to combine singular and plural words and manual clustering was used to identify groups of terms within the B list. Query expansion is then done manually using these groups to further identify literature related to these topics. Gordon & Lindsay were successful in replicating Swanson's Raynaud's/fish-oil [8] and migraine/magnesium [6] discoveries using this method. However, their method also involves a high degree of human involvement.

Gordon and Dumais[7] use Latent Semantic Indexing (LSI) as the underlying technique for their discovery methodology. The authors created a term by document matrix based on 560 documents in Medline that contained the term Raynaud's. LSI scaling was performed and the top 100 factors were selected. The cosine distance was measured from these terms to the term Raynaud's and a ranked list of B terms was formed according to decreasing order of cosines. From these B terms, the authors picked blood viscosity arbitrarily based on Swanson's discovery. LSI scaling was then performed again on Medline documents that contain blood viscosity and cosines were computed between the resulting list and Raynaud's. The authors hypothesized that this would show all the terms that were close to Raynaud's from blood viscosity's view point. However, the authors found that fish-oil was not highly ranked in this list even though their ranked B list was quite similar to the list that was obtained by Gordon and Lindsay.

Weeber [26] and Srinivasan [19] have been quite successful in replicating the Raynaud's/fish-oil discovery. They both employ Unified Medical Language Systems (UMLS)[1] concepts to reduce the size of the search space. Their method takes advantage of the semantic knowledge that is inherent in the UMLS. Weeber uses the Meta-map program to reduce the raw text in titles and abstracts to UMLS concepts. He further reduces the search space by what he calls semantic filtering where he collects all terms that come under a particular UMLS concept and then ranks them. Srinivasan uses the Medical Subject Headings (MeSH)[2] terms that have been indexed for titles and abstracts. These MeSH terms are then mapped to UMLS and weighted according to their occurrence frequency using TF*IDF scheme. An advantage of Srinivasan's method is that user is not required to intervene in the sorting of the lists. Employing MeSH together with UMLS ensures that the words that end up in the list do not contain words that may be considered irrelevant. This is the alternative to using the stop list that Swanson used to remove erroneous words. However this method relies heavily on external knowledge sources: the UMLS and MeSH.

Bruza et. al's method [1] addresses a variation of the open discovery process discussed earlier. This method involves compiling a set of literatures into a knowledge representation model known as the Hyperspace Analogue to Language (HAL). HAL represents words as vectors in a high dimensional space. An information flow metric is employed to discover implicit connections between the HAL vector for "Raynaud's" and other words, which are ranked in decreasing order of information flow from "Raynaud's". This method deviates significantly from the other authors in several aspects. Conceptually it tries to simulate how a human would come up with a hypothesis having read literatures on A and

[1] UMLS Knowledge Sources, 15th Edition, The National Library of Medicine.

[2] http://www.nlm.nih.gov/mesh/meshhome.html

C. Operationally it does not use a two step process of attaining the literatures. The literatures used in this method contains titles that do not contain any of A, B or C terms. This is also consistent with the way that a human would generally acquire knowledge. Nonetheless, a human is able to abduce new knowledge relating to concepts that they have read about in the past.

In summary, term co-occurrence information provides an important foundation for uncovering useful connections in the text. There seems to be no consensus as to how best exploit it. In this article, we adopt the position that cognitive knowledge representation in the computational form of semantic space naturally captures term co-occurrence information. Moreover, semantic space offers a flexible representation within which various weighting schemes can be examined. In this paper we take a "bare bones" approach by trying to replicate Swanson's Raynaud's/fish-oil and migraine-magnesium discoveries without recourse to external knowledge sources such as MeSH, or UMLS. This will provide some insight into the portability of text-based discovery in domains in which such knowledge sources are non existent, or less developed. More specifically, this paper explores the following questions:

1. Can term weighting schemes be used as a substitute for a carefully tailored and informed stop word list?
2. Is reduction of the semantic space beneficial for the open mode of discovery?

2.1 Semantic Space

If Swanson's discoveries were to be replicated automatically, what knowledge representation would be appropriate, and how could the hypothesis be generated? According to the philosopher C.S. Peirce, Swanson's explanatory hypothesis is a manifestation of abduction: "It [abduction] is the only logical operation which introduces any new idea; for induction does nothing but determine a value and deduction merely evolves the necessary consequences of a pure hypothesis" ([17], p216). Abduction has recently been considered from a psychologistic perspective which does not permit the reasoning process to be abstracted from the (human) agent performing the reasoning [5]. As abduction is a form of human reasoning, treating it from a psychologistic perspective is particularly apt.

A semantic space is a dimensional space in which words are represented as points, or vectors, in a high dimensional space. The meanings of the words are derived "..from the way words are used in a discourse context" [9]. A colloquial way of interpreting this view is the meaning of a word is determined by "the company it keeps".

There is a growing ensemble of semantic space models [15,3,13,14,10,11,16,12,18]. The most well known of these models in IR is Latent Semantic Indexing (LSI), which is known as Latent Semantic Analysis (LSA) in the cognitive science community. Even though there is ongoing debate about specific details of the respective models, they all feature quite a remarkable level of compatibility with a variety of human information processing tasks such as semantic word association. For this reason a semantic space would seem to be a promising basis on which to build a computational system which mimics the human's ability to form abductive associations between terms. Semantic space also offers a flexible representation within which various weighting schemes can be embedded. In a nutshell, semantic space is a single representation which reflects both statistical and "semantic" aspects.

Semantic spaces are built from text corpora using word co-occurrence information. As stated earlier, there seems no consensus in the literature about how best to use co-occurrence information. For this reason, and the cognitive track record mentioned above, semantic space would seem to be a particularly applicable model for investigating co-occurrence in relation to literature based discovery.

The basis of the semantic space used in the experiments reported below is an $n \times n$ term-term matrix denoted by S. The value of the cell $S[i, j]$ reflects the strength of co-occurrence of term i and term j. In the experiments reported below, co-occurrence is computed within the context of a Medline document title. For example, in a simple semantic space model, $S[i, j]$ is set to the number of documents in which both terms i and j co-occur. In the construction of a semantic space there is the tacit assumption that the frequency of co-occurrence of two words u and v gives some indication of the importance of v in establishing the meaning of u. In literature discovery however the value of frequency in establishing a connection between words is suspect. Highly frequent co-occurrences may be part of the background knowledge and therefore it may be the very infrequent co-occurrences that contain the surprises that convey useful information to the human. Therefore at the very least it is desirable to correct for the frequency bias inherent in semantic space models by term weighting.

2.2 Odds Ratio

Lowe [14] argues convincingly for the use of an odds ratio to compensate for the frequency bias inherent in co-occurrence counts. The odds ratio is a notion from statistics. We consider that Medline document titles are produced by a process of drawing word pairs from a bag. We are interested in comparing two situations: (i) the odds of getting a term u when we already have drawn a term v and (ii) the odds of getting u when we don't have v.

The *odds* of event, e, is the ratio of the probability of e to the probability of not e. When comparing two situations, s_1 and s_2 the odds ratio forms the ratio of the odds of e given s_1 and the odds of e given s_2.

Lowe estimates the ratio of the odds of drawing a pair containing u given that the pair contains v with the odds of drawing a pair containing u given that the pair does not contain v.

$$\theta = \frac{p(u|v)/p(\neg u|v)}{p(u|\neg v)/p(\neg u|\neg v)} \tag{1}$$

The probabilities in this function can be estimated from frequency counts. Before giving the formulae we introduce for notational convenience:

$$s(v) = \sum_i S[i, v] \quad \text{and} \quad ss = \sum_{i,j} S[i, j] \tag{2}$$

Given this notation the probabilities can be estimated by:

$$\hat{p}(u|v) = \frac{S[u,v]}{s(v)} \quad (3) \qquad \hat{p}(u|\neg v) = \frac{s(u) - S[u,v]}{ss - s(v)} \quad (5)$$

$$\hat{p}(\neg u|v) = \frac{s(v) - S[u,v]}{s(v)} \quad (4) \qquad \hat{p}(\neg u|\neg v) = \frac{ss - s(v) - s(u) + S[u,v]}{ss - s(v)} \quad (6)$$

In the odds ratio the denominators of the above expression cancel to leave

$$\hat{\theta}(u,v) = \frac{S[u,v](ss - s(v) - s(u) + S[u,v])}{(s(v) - S[u,v])(s(u) - S[u,v])} \tag{7}$$

Once θ has been estimated it is thresholded to be above 1.

$$\hat{\theta}_{th}(u,v) = \begin{cases} \hat{\theta}(u,v) & \text{if } \hat{\theta}(u,v) > 1 \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

Lowe reports that sometimes the log of the odds ratio is taken, however we found in our experiments that taking the log significantly impaired the performance of the odds ratio weighting scheme. In this paper we use the thresholded odds ratio to define the semantic space.

$$S[u,v] \leftarrow \theta_{th}(u,v) \tag{9}$$

2.3 Log Likelihood

The log-likelihood test introduced by Dunning[4] is a test for independence. The test attempts to measure whether or not the probability of two terms is independent or not, i.e. to test whether or not

$$Pr(uv) = Pr(u)Pr(v) \tag{10}$$

where $Pr(uv)$ is the probability that both the terms u and v will occur together in a document, $Pr(u)$ is the probability that u will occur in the document and so on.

The basic assumption is that documents are produced by a binomial process so that the likelihood of seeing the terms u and v together in k documents out of n is distributed according to the binomial distribution.

$$h(p_{uv}; k_{uv}, n) = p_{uv}^{k}(1 - p_{uv})^{n-k}\binom{n}{k} \tag{11}$$

Dunning's test is a log-likelihood test. It calculates the ratio of the maximum likelihood of the observed parameter given the independence assumption to the maximum likelihood of the observed parameters without the independence assumption

$$\lambda = \frac{\max\limits_{p_{uv}=p_u p_v} h(p_{uv}, p_u, p_v; k_{uv}, k_u, k_v, n)}{\max h(p_{uv}, p_u, p_v; k_{uv}, k_u, k_v, n)} \tag{12}$$

Table 1. Example HAL representation of the term "Raynaud"

nifedipine (0.44), scleroderma (0.36), ketanserin (0.22), synthetase (0.22), sclerosis (0.22), thromboxane (0.22), prostaglandin (0.22), dazoxobin (0.21), E1 (0.15), calcium (0.15), vasolidation (0.15), platelet (0.15), ..., platelets (0.07), blood (0.07), viscosity (0.07), vascular (0.07), ...

where p_{uv} is the probability of observing both u and v within a document and k_{uv} is the number of documents that contain both u and v, and n is total number of documents, and so on.

It is difficult to find an analytical or even computationally stable solution for Equation 12. To side-step this problem Dunning suggests the comparison of $p_{u|v}$ with $p_{u|\neg v}$ which leads to the analytically solvable:

$$\lambda = \frac{\max\limits_{p_{u|v}=p_{u|\neg v}} h(p_{u|v}, p_{u|\neg v}; k_{u|v}, k_{u|\neg v}, n_{u|v}, n_{u|\neg v})}{\max h(p_{u|v}, p_{u|\neg v}; k_{u|v}, k_{u|\neg v}, n_{u|v}, n_{u|\neg v})} \tag{13}$$

Solving this equation produces the following:

$$\lambda = \frac{h(p; k_{u|v}, n_{u|v}) h(p; k_{u|\neg v}, n_{u|\neg v})}{h(p_{u|v}; k_{u|v}, n_{u|v}) h(p_{u|\neg v}; k_{u|\neg v}, n)} \tag{14}$$

where $p = (k_{u|v} + k_{u|\neg v})/(n_{u|v} + n_{u|\neg v})$, $p_{u|v} = k_{u|v}/n_{u|v}$, and $p_{u|\neg v} = k_{u|\neg v}/n_{u|\neg v}$, $k_{u|v}$ is the number of documents containing both u and v, $k_{u|\neg v}$ is the number of documents containing u but not v, $n_{u|\neg v}$ is the number of documents containing v, and $n_{u|\neg v}$ is the number of document that don't contain v.

For both odds ratio and log-likelihood we only calculated their value if $k_{uv} > 0$ otherwise we set the value to zero. This was partly a pragmatic concern as to do otherwise generates very large matrices.

2.4 Hyperspace Analogue to Language

The Hyperspace Analogue to Language (HAL) model has quite remarkable success in replicating human semantic word association norms [3]. For this reason, it could be potentially effective for the discovery of B-terms. HAL has also used to replicate the Raynaud's/fish-oil discovery with some degree of success [1]. HAL weights co-occurrence via a linearly decreasing function of word distance within a context window of length l. Typically, $l = 10$. For reasons of brevity, the HAL algorithm is not discussed. Rather an example HAL vector for the term "Raynaud" is given in table 1 .

Dimensional Reduction of Semantic Space. Singular Value Decomposition (SVD), a theorem from linear algebra, projects the semantic space into a space of lower dimensionality with the effect that words of similar meaning tend to cluster. SVD has been used with encouraging effect to replicate human cognitive phenomena involving association or semantic similarity [10]. Its performance in replicating the Raynaud's/fish-oil discovery is disappointing [7]. Our pilot studies into using SVD for literature based discovery were also disappointing. Therefore, we adopt a different approach to reduce the space. The B-terms are integral to the discovery, so when a set of B-terms $\{B_1, \ldots, B_k\}$ are identified by the user, these terms are used to compute a semantic "subspace" defined by $\{B_1, \ldots, B_k\}$. More specifically, those titles are selected which contain at least one B-term B_i resulting in a subcorpus from which a semantic space is computed. Discovery of the $A - C$ connection is driven within this reduced space. This method of dimensional reduction is termed *contextual reduction.*

3 Discovery in Semantic Space

Recall from the introduction that the indirect $A - C$ connection is made via intermediate B-terms. If more of the B-terms are shared by the respective A and C vectors, the stronger the connection between A and C. The dot product of the A and C semantic space vectors (in both higher and lower dimensionality) is a means of computationally realizing this intuition provided the B-terms are prominently weighted in the representations of A and C. Alternatively, the cosine between A and C can be calculated in order to bridge the connection between A and C [7]. When A and C are both normalized to unit length, cosine equates to dot product.

Under the assumption that the B-terms are prominently weighted in the A and C representations, another computational means that can be brought to bear to establish the $A - C$ connection is to consider them as points and to measure the distance between them. However, when vectors are normalized to unit length the "relative ranking" achieved by cosine and Euclidean distance are the same [27]. For this reason, results are reported using cosine similarity.

Information flow computations through semantic space has shown some promise for computing suggestions relevant to the Raynaud's-fish-oil discovery[1] as well as computing related terms for automatic query expansion in text retrieval [2]. It is essentially an asymmetric form of dot product. Information flow thresholds the co-occurrence values with respect to two threshold values δ_1 and δ_2, and then forms a special type of vector product. The degree of information flow from u to v with respect to a thresholds δ_1 and δ_2 is given by:

$$\text{degree}_{\delta_1,\delta_2}(i,j) = \frac{\displaystyle\sum_{\substack{x \mid S[x,i]>\delta_1 \\ \text{and } S[x,j]>\delta_2}} S[x,i]}{\displaystyle\sum_{x \mid S[x,i]>\delta_1} S[x,i]} \qquad (15)$$

A high degree of information flow is achieved when many of the dimensions in the vector u above the threshold δ_1 are also present in vector v. It can therefore be considered as an heuristic form of dot product.

A common choice for δ_1 is the mean value of the non-zero components of the i vector, i.e

$$\delta_1 = \frac{1}{\text{count}(S[i,u] \neq 0)} \sum_i S[i,u] \tag{16}$$

The parameter δ_2 is commonly set to zero.

4 Replicating the Raynaud's/Fish-Oil Discovery

The experiments reported below are based on the two facets of the open discovery mode mentioned earlier:

1. Discovery of potential B-terms
2. Discovery of potential $A-$terms via the B-terms starting from C (i.e., making the indirect $A - C$ connection)

Stop words (normal vs. tailored), term weighting scheme (odds ratio, log-likelihood, HAL), and contextual reduction (with or without) were varied.

4.1 Data

A corpus of 111,603 Madeline core clinical journal articles from the period 1980-1985 were used. Only the titles of the articles were used as Swanson's Raynaud's/fish-oil discovery was made solely on the basis of document titles.

Two stop words lists were employed: A standard collection of stop words commonly used in IR experiments and the tailored stop words of the ARROWSMITH system.

4.2 Method

From these documents we then generated semantic spaces using (i) HAL with a window length of 10 and (ii) the simple semantic space model described earlier. The simple co-occurrence counts were additionally weighted using odds ratio θ and Dunning's log likelihood score $-2 \log \lambda$. Log-likelihood can produce non zero values for $k_{uv} = 0$, but these were excluded. The simple co-occurrence frequencies were generated using words without stemming. The resultant semantics spaces spanned 34716 dimensions (normal stop words removed), or 28779 dimensions if ARROWSMITH stop words were removed.

Method for Discovery of B-terms. The guiding intuition behind the discovery of B-terms follows Gordon & Lindsay's argument that the best B-terms are those that are semantically and statistically close to the $C-$term [8]. This view was supported by Gordon & Dumais, where a semantic neighbourhood was computed around the Raynaud vector using cosine in a SVD generated lower dimensional space. (The original higher space was a term-document matrix). We adopt a similar approach. A semantic space is computed as above and the semantic neighbourhood by ranking the terms on decreasing cosine with the Raynaud vector. Stop word lists and weighting schemes were manipulated covering all combinations. In the ensuing results, these rankings are referred to as a *semantic neighbourhood ranking.* Only those terms which explicitly co-occur with "Raynaud" were ranked reflecting the intuition in the literature that B-terms are related explicitly to the phenomenon represented by C.

In addition, the rankings of terms within the Raynaud vector itself were examined (see table 1). As with the semantic neighbourhood rankings, stop word list and weighting scheme were manipulated. These rankings are referred to as *within vector term rankings* as the terms with positive value in the Raynaud vector are terms which appeared in the same context(s) as the term "Raynaud".

Rankings were evaluated by examining the rank of known B-terms: "platelet", "blood "viscosity" and "vascular". (These are single terms reflecting the B-terms "platelet aggregation", "blood viscosity" and "vascular reactivity" mentioned in the introduction). The higher that these terms were ranked, the better the result. Both the actual rank, and the percentage distance from the top of ranking are reported.

Method for $A - C$ Discovery. Once a set of $B-$terms has been discovered, they can be used to prime an $A-C$ discovery in the following way. The C vector (corresponding to "Raynaud") is normalized to unit length. Thereafter, those dimensions corresponding to the $B-$terms are given a maximal weight of one. This corresponds to the situation where the scientist is giving positive relevance feedback with respect to those terms (s)he assesses as salient to addressing the phenomenon represented by C. In the experiments reported below, the known $B-$terms *blood*, *viscosity*, *vascular*, and *platelet* were boosted manually. Cosine and information flow are measured to all terms in the vocabulary, and ranked. The ranks of the terms "fish" and "oil" are then inspected. Both the actual rank, and the percentage distance from the top of ranking are reported.

4.3 Results

For reasons of brevity only selected results are reported.

Discovery of B-terms. Table 2(a) and table 3 depict the effect of stop word list and weighting scheme on the ranking of $B-$terms in higher space.

For both within vector ranking and semantic neighbourhood, log-likelihood appears to be superior for both normal stop words and the ARROWSMITH stop words. It outperforms odds ratio possibly because odds ratio favours infrequently occurring terms. The linearly decreasing function used in HAL does not seem to promote the discovery of B-terms as effectively as log-likelihood. This may be evidence of HAL being subject to frequency bias [14].

Ranking of B-terms using the ARROWSMITH stop words (table 2(a)) is only marginally better than with normal stop words. This is an encouraging result, as manually crafted stop words are time-consuming to produce.

Interestingly the best within vector ranking (LL columns in table 2(a)) is very similar to the best semantic neighbourhood ranking for both normal and ARROWSMITH stop words. Both of these rankings were produced by log-likelihood. This suggests that computing the semantic neighbourhood maybe

Table 2. A comparison of weighting schemes and stop word lists for (a) "within Raynaud term ranking", and (b) "within migraine term ranking"

B-Term	Normal			Arrowsmith		
	HAL	LL	Odds	HAL	LL	Odds
blood	76	43	166	47	12	61
viscosity	54	43	25	26	12	19
platelet	37	36	115	10	11	52
vascular	77	43	125	27	12	54
Average	61	**41**	107	27	**11**	46

(a)

B-Term	HAL	LL	Odds
platelet	21	64	780
aggregattion	201	130	703
ischemia	44	17	563
prostaglandins	1237	645	953
sodium	56	294	1110
calcium	195	382	1205
channel	201	155	581
cerebrospinal	573	411	991
blocker	307	69	456
Average	315	**240**	815

(b)

Table 3. A comparison of weighting schemes and stop word lists for the cosine semantic neighbourhood of "Raynaud"

	blood	viscosity	platelet	vascular
Standard Stop words				
- Log-likelihood	43 (0.12%)	43 (0.12%)	28 (0.08%)	43 (0.12%)
- Odds-ratio	166 (0.48%)	81 (0.23%)	76 (0.22%)	144 (0.41%)
- HAL	127 (0.37%)	140 (0.40%)	56 (0.16%)	9 (0.03%)
ARROWSMITH Stop words				
- Log-likelihood	11 (0.04%)	11 (0.04%)	7 (0.02%)	11 (0.04%)
- Odds-ratio	34 (0.12%)	39 (0.14%)	32 (0.11%)	46 (0.16%)
- HAL	37 (0.13%)	40 (0.14%)	6 (0.02%)	12 (0.04%)

redundant. This would be a significant saving in computational cost as some discoveries may potentially involve huge numbers of terms. Effective rankings of B-terms appear to arise as a product of the construction of the semantic space with log-likelihood weighting.

Discovery of $A - C$ Connections. Table 4(a) depicts the best performing results for replicating the Raynaud's/fish-oil discovery in both the high-dimensional semantic space and the contextually-reduced space. The striking feature is the excellent performance of information flow across stop words when primed with odds ratio weights. The performance of information flow in a contextually reduced space is ideal with both fish and oil at the head of the ranking. This is significant as driving discovery within a reduced space can be orders of magnitude more efficient than in the higher space.

Cosine's performance lags considerably behind that of information flow in both high and lower space. This may be due to the fact that information flow only considers those dimensions above average weight in the Raynaud vector, thereby better "focussing" for the purposes of discovery.

Log-likelihood does not perform well in the discovery of $A - C$ connections. This is in contrast to its good performance in the the discovery of B-terms. It is

Table 4. Similarity to A-terms in semantic space after boosting B-term weights in (a) the Raynaud vector, and (b) the migraine vector

	Info. Flow		**Cosine**	
	fish	oil	fish	oil
High				
Normal (Odds.)	5	5	197	445
Arr. (Odds.)	4	4	346	384
Low				
Normal (Odds.)	1	1	49	25
Arr. (Odds.)	1	1	82	77
Arr. (None)	3	3	39	12

(a)

	Info. Flow	**Cosine**
	magnes.	magnes.
High		
Normal (Odds.)	72	612
Normal (Simple)	707	8
Low		
Normal (Simple)	296	573

(b)

hard to fathom this mix of results. We speculate that odds ratio are more stable in relation to the manual boosting of B-terms, than is log-likelihood.

5 Replicating the Migraine-Magnesium Discovery

The replication of the Raynaud's/fish-oil discovery showed log-likelihood weighting as beneficial for the discovery of B-terms and odds ratio weighting beneficial for the discovery of the $A - C$ connection, particularly with information flow in both higher and contextually reduced space. In both cases, a tailored stop word list seems not to be necessary. In order to investigate the potential of these results to carry over to other discoveries, the following experiment replicates the migraine-magnesium discovery within the same experimental framework as used for the Raynaud's/fish-oil discovery. A semantic space was computed from the same corpus as the previous experiment, but abstracts were included to examine the effect of extra information. The resulting semantic space comprised 90,023 dimensions and the average document length is significantly greater than when considering titles alone. The standard stop word list was used.

5.1 Discovery of B-terms

Table 2(b) shows that log-likelihood is once again the best weighting scheme for the promotion of B-terms in terms of average B-term ranking. There is a cautionary note, however: in the migraine/magnesium discovery not all of the B-terms had their rank improved log-likelihood. Log-likelihood is a test of dependence, and some terms such as "platelet" while having a high rank according to HAL which is frequency based, have a somewhat lower rank according to log-likelihood.

5.2 Discovery of $A - C$ Connections

Table 4 shows the best performing results for information flow and cosine. In this case, cosine is clearly superior and performance of both information flow and cosine within a contextually reduced space is disappointing. This may because the reduced space is considerably larger than was the case when only titles were used as the basis of the contextually reduced subspace. As a consequence, many more associations are represented and effectively "muddying" the discovery process.

6 Summary and Conclusions

This paper analyzes the effect of term weighting, stop word list and dimensional reduction on literature-based discovery. Swanson's Raynaud's/fish-oil and migraine-magnesium discoveries were used as case studies. Even though we cannot claim these case studies as being typical, we feel that some of their characteristics would carry across to other discoveries. Two aspects of discovery were brought under focus. The discovery of intermediate, or B-terms, and the discovery of indirect $A - C$ connections terms via these B-terms. As Swanson's scientific discoveries are examples of abduction, a form of human reasoning, a cognitively motivated form of knowledge representation was employed called semantic space. Semantic space provides representations of words in a dimensional space. Moreover it allows both semantic and statistical issues to be reflected in the one representation framework.

This paper set out to address whether a bare bones approach to literature-based discovery is possible. This goal is significant because most successful replications of discoveries have involved significant manual interventions and/or the use of external knolwedge sources. In relation to the question whether weighting schemes can be used as a substitute for tailored stop word lists, there is evidence toward the affirmative, for both discoveries based on titles, or titles plus abstracts. This is encouraging as tailored stop word lists are time consuming to create and they need to be constructed *per* discovery. In regard to the question whether dimensional reduction is beneficial to discovery, the evidence from our studies is ambivalent when contextual reduction of the space is performed. More research is warranted to investigate this issue further as driving discoveries in reduced space has the potential to be far more efficient than in the higher dimensional semantic space. More specific conclusions are as follows:

- Log-likelihood weighting appears to promote discovery of B-terms to an encouraging degree. It should seriously be considered as part of the arsenal for the discovery of B-terms mindful that it may not promote all salient B-terms.
- If a significant set of relevant B-terms have been discovered, $A - C$ connections can be discovered to an encouraging degree by computations in semantic space. Both information flow (an asymmetric measure) and a similarity metric such as cosine can be used. Cosine is more "forgiving" than information flow and can best better be employed when the user is not fully confident of the associated B-term set, or when discovery is performed over both titles and abstracts. More research is needed to understand their relative strengths and weaknesses of cosine and information flow.

We envisage literature based discovery to be an interactive process somewhat akin to relevance feedback in IR. The scientist can first browse suggestions for

B-terms from the system, use his or her background knowledge to enhance the suggestions, or change their weights. These can then be used to compute A-C suggestions with the A and/or C vector representations conditioned by feedback from the scientist, e.g., boosting weights where prominent. Statistical weighting in semantic space provides only part of the machinery for literature based discovery. Future work needs to consider how best to gather and incorporate relevance feedback from the scientist, or user. Clearly, single terms are problematic as they can be ambiguous. Phrases would perhaps provide a better vehicle for harnessing feedback. If so, research is needed to efficiently enhance semantic space representation with phrases.

Acknowledgments

The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Education, Science, and Training).

References

1. P. Bruza, D. Song, and R. McArthur. Abduction in semantic space: Towards a logic of discovery. *Logic Journal of the Interest Group in Pure and Applied Logics*, 12:97–109, 2004.
2. P.D. Bruza and D. Song. Inferring Query Models by Computing Information Flow. In *Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM 2002)*, pages 260–269. ACM Press, 2002.
3. C. Burgess, K. Livesay, and K. Lund. Explorations in context space: words, sentences, discourse. *Discourse Processes*, 25(2&3):211–257, 1998.
4. Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1994.
5. D. Gabbay and J. Woods. *The Reach of Abduction: Insight and Trial*, volume 2 of *A Practical Logic of Cognitive Systems*. Elsevier, 2004. An early draft appeared as Lecture Notes from ESSLLI 2000 (European Summer School on Logic, Language and Information), Online: http://www.cs.bham.ac.uk/ esslli/notes/gabbay.html.
6. M. D. Gordon. Literature-based discovery by lexical statistics. *Journal of the American Society for Information Science*, 50:574–587, 1999.
7. M. D. Gordon and S. Dumais. Using latent semantic indexing for literature based discovery. *Journal of the American Society for Information Science*, 48:674–685, 1998.
8. M. D. Gordon and R. L. Lindsay. Towards discovery support systems: A replication, re-examination, and extension of swanson's work on literature-based discovery of a connection between raynaud's and fish oil. *Journal of the American Society for Information Science*, 47:116–128, 1996.
9. W Kintsch. Predication. *Cognitive Science*, 25:173–202, 2001.
10. T. K. Landauer and S. T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240, 1997.
11. T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

12. J. P. Levy and J. A. Bulliniaria. Learning lexical properties from word usage patterns: Which context words should be used? *Connectionist models of learning, development and evolution*, pages 213–282, 1999.
13. W. Lowe. What is the dimensionality of human semantic space? In *Proceedings of the 6th Neural Computation and Psychology workshop*, pages 303–311. Springer Verlag, 2000.
14. W. Lowe. Towards a theory of semantic space. In J. D. Moore and K. Stenning, editors, *Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society*, pages 576–581. Lawrence Erlbaum Associates, 2001.
15. K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments & Computers*, 28(2):203–208, 1996.
16. M. Patel, J. A. Bulliniaria, and J. P. Levy. Extracting semantic representations from large text corpora. *Proceedings of the Fourth Neural Computation and Psychology Workshop*, pages 199–212, 1997.
17. C.S Peirce. The Nature of Meaning. In Peirce Edition Project, editor, *Essential Peirce: Selected Philosophical Writings Vol 2 (1893-1913)*, pages 208–225. Indiana Univ. Press, 1998.
18. M. Sahlgren. Towards a flexible model of word meaning. In *Proceedings of AAAI Spring Symposium 2002*, Palo Alto, California, USA, 2002. Stanford University.
19. P. Srinivasan. Text mining: Generating hypotheses from medline. *Journal of the American Society for Information Science and Technology*, 55(5):396–413, March 2004.
20. D. R. Swanson. Fish oil, raynaud's syndrome, and undiscovered public knowledge. *Perspectives in Biology and Medicine*, 30:7–18, 1986.
21. D. R. Swanson. Undiscovered public knowledge. *Library Quarterly*, 56:103–118, 1986.
22. D. R. Swanson. Two medical literatures that are logically but not bibliographically connected. *Journal of the American Society for Information Science*, 38:228–233, 1987.
23. D. R. Swanson and N. R. Smalheiser. Implicit text linkages between medline records: Using arrowsmith as an aid to scientific discovery. *Library Trends*, 48:48–59, 1999.
24. D.R. Swanson and N.R. Smalheiser. An interactive system for finding complementary literatures: A stimulus to scientific discovery. *Artificial Intelligence*, 91(2):183–203, 1997.
25. M. Weeber, R. Vos, H. Klein, and de Jong-van den Berg. Using concepts in literature-based discovery: simulating swanson's raynaud- fish-oil and migraine-magnesium discoveries. *JASIST*, 52(7):548–557, 2001.
26. Marc Weeber, Henny Klein, and Lolkje T. W. Jong-can den Berg. Using concepts in literature-based discovery: Simulating swanson's raynaud-fish oil and migrain-magnesium discoveries. *Journal of the American Society for Information Science and Technology*, 52(7):548–557, 2001.
27. D. Widdows. *Geometry and Meaning.* CSLI Publications, 2004.

Assisting Scientific Discovery with an Adaptive Problem Solver

Christopher Dartnell[1] and Jean Sallantin[2]

[1] Euriware, Octeville,
44 Rue des Vindits, 50130 Cherbourg-Octeville - France
christopher.dartnell@euriware.fr
[2] LIRMM, UMR 5506,
161 rue Ada, 34392 Montpellier Cedex 5 - France
js@lirmm.fr

Abstract. This paper is an attempt to design an interaction protocol for a multi-agent learning platform to assist a human community in their task of scientific discovery. Designing tools to assist Scientific Discovery offers a challenging problematic, since the problems studied by scientists are not yet solved, and valid models are not yet available. It is therefore impossible to create a problem solver to simulate a given phenomenon and explain or predict facts. We propose to assist scientists with learning machines considered as adaptive problem solvers, to build interactively a consistent model suited for reasoning, simulating, predicting, and explaining facts.

The interaction protocol presented in this paper is based on Angluin's "Learning from Different Teachers" [1] and we extend the original protocol to make it operational to assist scientists solve open problems. The main problem we deal with is that this learning model supposes the existence of teachers having previously solved the problem. These teachers are able to answer the learner's queries whereas this is not the case in the context of Scientific Discovery in which it is only possible to refute a model by finding experimental processes revealing contradictions. Our first contribution is to directly use Angluin's interaction protocol to let a machine learn a program that approximates the theory of a scientist, and to help him improve this theory. Our second contribution is to attenuate Angluin's protocol to take into account a social cognition level during which multiple scientists interact with each other by the means of publications and refutations of rival theories. The program learned by the machine can be included in a publication to avoid false refutations coming from a wrong interpretation of the theory.

1 Introduction

Assistance to Scientific Discovery is a very challenging research domain: scientists study open problems which have never been solved. Therefore, no satisfying model or theory might already exist, so we propose to assist scientists by learning machines in their task of theory building. Such a software assistant can learn to

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 99–112, 2005.

simulate an observed phenomenon explain or predict facts, has to deal with uncertainty, pattern discovery, interactive ontology building [2], and has to produce statements comprehensible to a human to improve human-machine interaction. We divide the problem solving process in 3 steps:

1. The user, acting as the teacher, interacts with his learning assistant to make him learn his hypothesis. To model the interaction between a scientist validating his hypothesis with his assistant, we directly exploit Dana Angluin's Learning from different teachers paradigm [1] which formalizes a protocol for a human compliant robust learning defined as the result of a stable interaction cycle between a Learner and a Teacher. Her conclusions are completed by theoretical results about query driven machine learning complexity in [3] and [4].
2. The assistant brings a critical attitude concerning an approximation of the user's hypothesis to confirm or invalidate his hypothesis, and this interaction can lead to a revision of the hypothesis and/or of the description model, in which case they are both considered as learners. The user, at the heart of the system, builds interactively with an adaptive problem solver an adequate description model of the studied phenomena: he is in charge of providing a description model, and the adaptive problem solver uses machine learning and paraconsistent logic to detect contradictions between the learned theories and empirical results, or inadequateness between the description model and learned theories. These contradictions are used to initiate an Angluin-like interaction cycle during which the user learns at the same time as the machine, and this co-learning leads to a pertinent understanding of the problem.
3. Once the user considers that the approximation of his theory learned by his assistant is expressive enough, he can use it to publish his own theory: each theory proposed by a scientist is not refutable, but a logical theory produced by a machine can always be reduced to a universal form which is refutable by an existential statement. Our contribution is to extend Angluin's protocol by introducing a social interaction level inspired by Popper's philosophy of science [5] and based on proofs and refutations of publications. The publications are logical conjectures which have to be submitted to the judgment of other learners to be pitilessly tested, put into question, and eventually falsified.

In section 2, we define the needed functionalities that a problem solver should implement to be adaptive and autonomous, and we emphasize that such an adaptive problem solver has to reason in paraconsistent logic to cope with contradictions. We show in section 3 that these contradictions are at the source of the interaction between the solver and the human it's assisting, and how this interaction is formalized in [1] by the use of *Membership and equivalence queries*. However, this learning model supposes an access to a Teacher to answer these queries whereas there isn't any to help scientists understand Nature and its laws, so we propose in sections 3.2 and 3.3 two extensions of this model to make it operational in the context of scientific discovery, and we validate in section 4 this protocol on a toy game, E+N. Finally, we present some experimental results before concluding.

2 Toward a Definition of an Adaptive Problem Solver

Common definitions of a problem solver take into account the type of solvable problem which characterizes it, as a differential equation problem solver, or nonlinear equation systems solver: a problem solver is designed to perform the computation of a known problem which has already been solved and modelled. So for any presented instantiation of the specific problem, it is able to solve it and produce its solutions.

An adaptive and autonomous problem solver should be able to acquire new abilities by learning how to solve new problems, and use this knowledge and experience to find solutions. To solve an open problem, one has to observe the problematic situation and analyse it to build a language describing the situation's dimensions pertinent for reasoning. These dimensions determine the definition domain of the variables characterising the problem and influencing the solution's computation. The language thus defined is used to formulate assumptions and hypothesis that have to be experimented. Comparing empirical results and theoretical computations can reveal contradictions between a theory and reality, and therefore lead to a revision of the description model and to the formulation of new hypothesis.

By making an analogy with the process of scientific discovery, in which neither the *ontology* nor the *theory* are perfectly known *a-priori*, we define below the functionalities that an adaptive autonomous problem solver should be empowered with to assist the process of discovery. It should be able to build and maintain an *Ontology* of the domain. By *Ontology*, we mean a logical language relevant with observations describing the pertinent dimensions of the problem, i.e. the types of the variables involved in its resolution. Furthermore, we want the ontology building to be the consequence of the interactive learning process of the logical language.

The principles of nominalization and reducibility [6] are the keys of a problem solver's adaptability, since they allow it to manipulate new concepts and design experimentations to validate the pertinence of these new dimensions for the computation of the problem's solutions:

- The solver should be able to learn ontological statements to constraint the relations between the values of the problem's dimensions, by analysing and correlating gathered information.
- The solver should be able to theorize: discover, name, and symbolically use regularities or patterns appearing on data by revising the ontology and introducing new dimensions to the problem's formulation. Transforming an observed property into a symbolic object and re-use it is called the *Nominalization* principle. This principle is essential to formulate and express a *theory* to explain the problem and predict further results. By *theory*, we mean a set of rules used to compute a problem's solutions.

- The solver should be able to empirically validate theories: transcribing mathematical abstractions to design experimentations feasible in the real world is called the *Reducibility* principle.

Interactions between the solver and its environment are *sine qua non* conditions of its evolution: by comparing the results of theoretical computations and the results of its interactions with the environment, the solver is able to detect contradictions in the formulated theories. These contradictions are used to motivate the actions and reflections of the adaptive problem solver: each experimentation is made to validate a theory, and is preceded by a prediction about its consequences. This prediction is compared to observed results to search for contradictions. Of course, the most informing situation is when a contradiction is detected, because it reveals either a wrong formulation of the problem by the user (perhaps a parameter was forgotten), or a inconsistency in the learned theory (coming from a bias in the learning set). To reason in the presence of contradictions, the logical ontology must be paraconsistent [7]: paraconsistent logics don't allow absurd reasoning (*ex-contradiction sequitur quod libet*), i.e. a statement and its negation can be true at the same time.

The following deduction shows that the paraconsistent contradiction principle requires four arguments to deduce a contradiction about A:

$$\frac{\neg A \vdash B \quad \neg A \vdash \neg B \quad \neg A \quad \vdash \neg(B \wedge \neg B)}{A \wedge \neg A}$$

$$\frac{\neg pope \vdash rain \quad \neg pope \vdash \neg rain \quad \neg pope \quad \vdash \neg(rain \wedge \neg rain)}{pope \wedge \neg pope}$$

In this example, all the arguments are evaluated. A contradiction Only if the contradiction "$\vdash \neg(rain \wedge \neg rain)$" is not admitted, then “it is paradoxical to be pope”.

This is very useful when reasoning on descriptions coming from different contexts or points of view, and [8] gives an elegant example of paraconsistency based on a defeasible deontic logic:

- a *Paraconsistent Logic* allows to reason in presence of contradictions in order to maintain obligations.
- a *Deontic Logic* allows to maintain a past knowledge by using the *Obligation*, *Forbiddance* and *Permission* modalities. In the context of machine learning, an obligatory fact is a fact which provoques a major contradiction when false, a forbiden fact is a fact which provoque a major contradiction when true. An advised (not obligatory) fact will provoque a minor contradiction when false, a disadvised fact (not forbiden) will provoque a minor contradiction when true.
- a *Defeasible Logic* allows to revise the model when new contradictory facts occur, and to produce a new theory adapting the strength of the contradictions.

Deontic logic is used to localise contradictions and provoke a revision in the set of defeasible theories, and this paraconsistency allows the solver to adapt the ontology to new facts and new observations.

Paraconsistent defeasible deontic logic rules describing a complex system are not easy to determinate, and since this kind of monotonous and multivaluated logics have experimentaly been shown as learnable using Angluin's paradigm [2], our goal is to have them learned by an adaptive problem solver interacting with a human.

In the following section, we discuss how Angluin's interaction protocol for machine learning can be used to formalise the necessary interactions between such a solver and its environment. We use contradictions to drive this interaction. We also propose an extension of this protocol to adapt concept learning theory to scientific discovery.

3 Making Angluin's Formalism Operational in the Context of Scientific Discovery

Angluin's formalism [1] gives a strong basis to interactive learning from different teachers, and introduces the idea that a learner could possibly become a teacher for another learner. We present this formalism in section 3.1, then we apply this protocol to assist scientific discovery with a learning machine in section 3.2. We show in section 3.3 how we introduce a social interaction level between learners to make this protocol operational in the context of scientific discovery, *i.e.* to cope with the apparent impossibility to use Equivalence queries.

3.1 Formal Aspects

Formal learning models differ by the information sources, by *a priori* knowledge given to the Learner, by its tasks and abilities, and by the success criteria of the learning process. In the model of exact identification with queries, studied in [3] and [4], the task is to identify an unknown concept drawn from a known concept class using queries to gather information about the unknown concept.

The interest of Angluin's works lies in the theoretic results she provides about the learnability of different concept classes (as monotonous DNF which are not learnable in the case of PAC learning or online learning) by methods based on the use and the combination of two main types of queries: *Membership* and *Equivalence* queries defined as follows:

Let the *domain* X be a nonempty finite set. A *concept* c is a subset of X, and a *concept class* C is any nonempty set of concepts. In a *Membership query* (MQ), the learner exhibits an example $x \in X$, and the access to an oracle returns 1 if $x \in c$, and 0 if $x \notin c$. In an *Equivalence query* (EQ), the learner exhibits a concept $c' \subseteq X$, and the oracle's answer is either "yes" if $c' = c$, or an element x in the symmetric difference of c and c', if $c' \neq c$. In [9] and [4], Angluin demonstrates the necessity of combining MQs and EQs to allow a powerful and effective learning. [1] formalizes a learning model based on the interaction between a *Learner* L and a *Teacher* T. Both of them are modelled as computers, and T is assumed to have a program p representing the concept to be taught to the learner, as illustrated in figure 1.

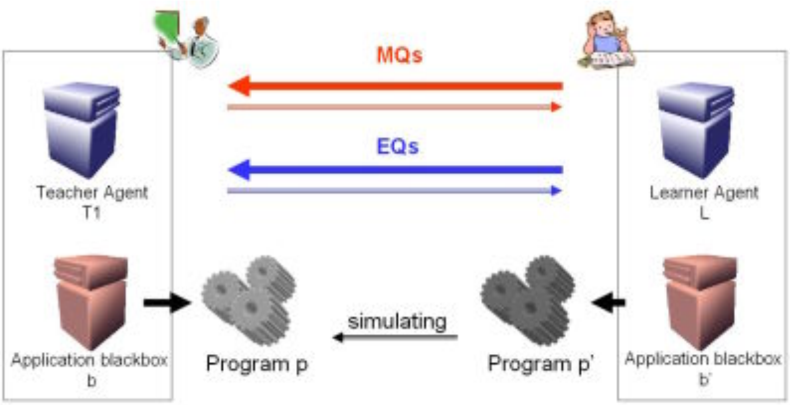


Fig. 1. Exact identification with queries

The teaching protocol involves examples of the concept, and possibly *other information* (we bring up this topic again later in this section) from which the learner is to develop a program p' that also represents the target concept. [3] emphasizes the fact that outright coding, in which T would transmit (using an encoding via examples) the text of the program p to L, is neither compliant, nor representative of human learning. Indeed, the "hardware and software environment" differs quite substantially from one person to another. In other words, all of us don't have the same brains, nor the same ways of thinking, although our anatomies are comparable.

Angluin illustrates this point of view with the human learning of juggling, which brings in muscular and visual reflexes, time perception and so on... These are "low level" mechanisms from which we only know very few about their triggering and their control. We merely know how to use or how to interpret their inputs and outputs, doing so in a symbolic way. The idea we want to put forward is that *to realize a task, acquire an ability, or identify a concept, the learner has to learn how to correctly use and combine "black boxes" representing the mechanisms triggered during the execution of the task, which are only partly and poorly known to him.* This example gave us the motivation to test this protocol in the context of scientific discovery (see section 3.3): according to Popper's conception of Scientific Discovery [5], scientists try to understand Nature's laws, by designing experiments and formulating theories on the basis of their results.

We show in section 3.2 and 3.3 how Angluin's protocol, in such a context, can be used at different levels of interaction.

Here is the theorem enunciated in [1]: *There exists a learner L* such that for every total recursive function b(x, s) there is a teacher T * b such that for every universal function box g' and every function box g that is b-related to g', L*(g) learns all the partial recursive functions from T * b (g'). Furthermore, L* is box-and-teacher-proof.* Using works as [10], [1] demonstrates the identification in the limit [11] of this process.

This means that *whatever might T's and L's applications black box be (applications being formalized by Angluin by recursive functions), L will learn after a finite number of queries, a program p' simulating p and producing only a finite number of errors if:*

1. *The computing performances of T and L are comparable, which means L is "not too slow" compared to T,*
2. *T has already managed to solve the problem.*

This theorem stands in the context of language learning, to which the context of scientific discovery is comparable since scientists aim at learning or discovering a language adapted to describe their environment and various phenomena occurring in it.

This protocol is clear and simple, and it ensures the convergence of the learning process, or at least, it ensures that whoever might the teachers be, the learner will not converge towards an incorrect solution. An adaptive problem solver can then learn a theory formulated by a scientist during the process of scientific discovery. However, in the context of scientific discovery, the Nature, which is considered as the teacher, is "silent" and cannot answer all learner's queries: the learner may still use MQs, by designing experiments and interpreting their results, but there is no way he can access Nature to answer his EQs ("is earth flat?", "is the law of gravitation true?").

We show in section 3.2 how the learning assistant can bring a critical point of view to the scientist while analysing experimentations' results and formulating theories, and in section 3.3 how we extend the model "Learning from Different Teachers" to "Learning from each other" by introducing multiple learners and interactions between them to confront a learner's interpretation of Nature with others'.

3.2 Interactive Aspects: Individual Reasoning

A scientist L learns from Nature T, by experimenting his hypothesis. In our approach of assistance to scientific discovery, we want the scientist to interact with an adaptive problem solver to find the solution of a problem, and the solution comes from the co-learning of these two entities.

An intelligent assistant is an adaptive problem solver, as described in section 2, able to analyse facts described in the language of the ontology written by the researcher while observing and describing the problem. The intelligent assistant that we develop [12] uses induction and abduction methods coming from machine learning with graphs and Galois lattice theory [13] which allow to find relevant logical implications and equivalence rules between the descriptors introduced by the user to describe the facts observed (see fig. 2). These rules can be easily understood by the researcher since they are formulated with his own words. The assistant can then induce theories predicting the behaviour of the studied system, and use abduction to explain past facts and to design experimentations testing the validity of the produced logical rules.

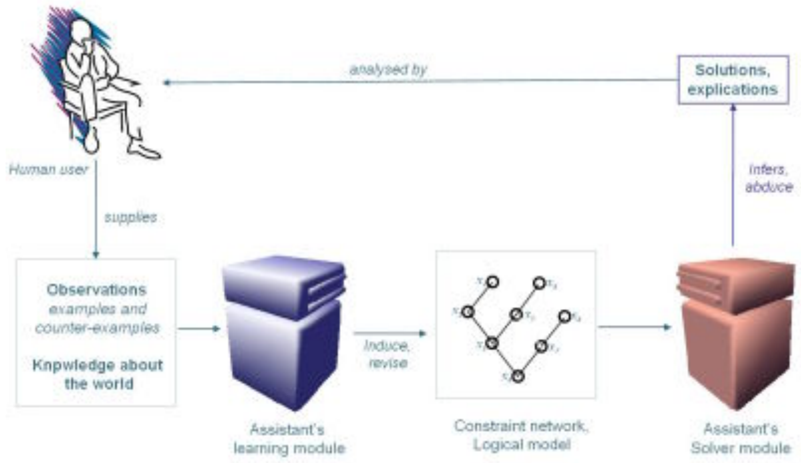

analysed by
Solutions, explications
Human user
supplies
Infers, abduce
Observations
examples and counter-examples
Knpwledge about the world
Induce, revise
Assistant's learning module
Constraint network, Logical model
Assistant's Solver module

Fig. 2. Human-Machine interaction cycle

Compared to Angluin's protocol, we propose to let the Learner be a couple of Learners: a scientist and his assistant, learning as well from their common Teacher (Nature) than from each other. The interaction between them follows the protocol presented in section 3.1, with this difference that both entities can act in turn either as the Teacher or as the Learner. So the scientist is in charge of formulating a description model of the problem's domain, and to modify it when irrelevant examples arise: erroneous predictions invalidate either the theories, either the initial conditions (description model). Therefore, he represents the teacher guiding the learning machine, and can also eliminate learning errors coming from bias of the learning set, for example by designing experiments to produce results considered by him as informant. The assistant can analyse large data and formulate an opinion concerning the scientist's choices, and act as the Teacher by anticipating negative result of MQs or EQs.

This justifies the use of a paraconsistent defeasible deontic logic to localise contradictions in the scientist's interpretation of results, in the approximation of his theory learned by the assistant, or in the discretization of the problem by the scientist, *i.e.* in his description model. We saw how Angluin's protocol formalized the interaction between a scientist and his assistant, and how the link to their common Teacher was made by designing experimentations and interpreting their results. The next section deals with the need of a social game between learners to answer one's EQs.

3.3 Social Aspects: Collective Cognition

As we introduced in section 3.1, the Nature, which is considered as the Teacher from which scientists learn during the process of scientific discovery, cannot be accessed to answer EQs, so we introduced a social interaction level to answer

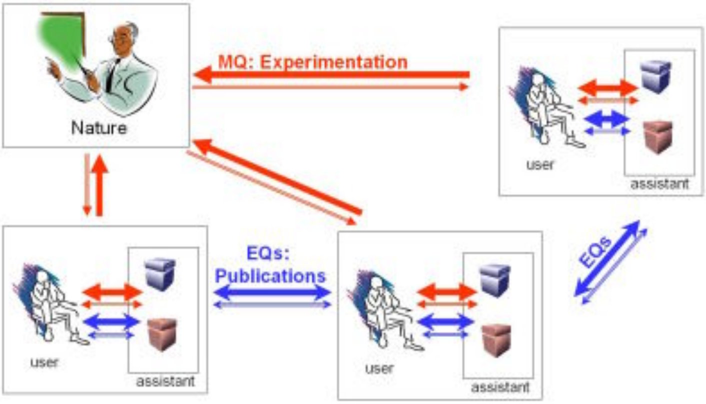


Fig. 3. Coping with EQs without an oracle

these queries. A scientist is member of a community, and published theories are temporary solutions accepted until they become insufficient to explain Nature: in our model, learners (who are couples of scientists and their assistant) are confronted to the judgment of other learners to cope with the impossibility to access an oracle for EQs in the context of scientific discovery. Every learner has the same access to Nature for MQs, whether they have proper interpretations and points of view, and they are in charge of answering other's EQs, as shown on fig.3.

Angluin's prospect of letting a learner becoming a teacher is meaningful for us, and our model let other learners answer EQs by the means of publications and refutations. Doing so, we allow the learners to act on behalf of a teacher by refuting other's hypothesis. According to "Learning From Different Teachers" theory, learning is still possible in these conditions (if the learners are *teacher-proof*). We symbolize the product of this social interaction by a score and a profit function. By attributing or deducing points for each query, depending on the oracle's answer, we can create a competition atmosphere or collaborative work between multiple learners. This atmosphere motivates the emission of EQs to score points and MQs to prove or refute a theory. The introduction of this social level can lead to experiment different points attribution in order to determine in which condition the community formed by the learners converges faster to an acceptable solution. These kinds of experiments are planned by cognitive scientists [14], and some of them have already taken place with human players only.

We shall now describe these experimentations with the game *E+N* and link the manipulated concepts with the notions presented previously.

4 Protocol Validation on a Toy Game: $E + N$

We implemented this toy game E+N to validate the protocol of assistance to scientific discovery presented in previous sections. In this experimentation, we aim at defining the limits of our protocol, and having a standard reference to evaluate further experiments implying scientists and their assistants.

4.1 Problem's Definition: Eleusis

The problem in Abbott's Eleusis card game [15] is to find a secret law hidden from the players and determining the valid sequences of cards that can be played during the game. The difficulty of the game can be adapted by:

- changing the length of the sequences concerned by the secret rule, to increase the complexity of the learning problem.
- fixing the choices offered by the rule, determining the ramifications in the resolution space. This might lead to formulations of various classes of Boolean formulas, as *CNF, DNF, k-term-DNF, ...*
- giving or hiding this information to players. This allows letting the learner fix his own learning biases or not.
- providing or not the Ontology used to explain the rule. This might be equivalent to concept learning on a finite or infinite domain

Players can formulate membership queries (MQs) by proposing a sequence of cards which is accepted or rejected by an oracle machine simulating Nature, and build on the basis of their experiments a theory consistent with their current knowledge to explain the hidden rule and predict further sequences. Since a concept learning problem can be assimilated to the problem of learning the mapping function between a set of examples ($x \in X$, X being a non empty finite set) and the Boolean value representing the belonging of x to the unknown concept $c \subseteq X$, we assume that it is suited to apply concept learning theory and use the interaction protocol formalized by [1].

Experimentations are in fact membership queries ("is $g(x)$ true?", g being an hypothesis, is an MQ), with this difference that experimentations often have a cost (time, resource, ...). [1] showed that algorithms using only membership queries were less performing than algorithms using membership queries combined with equivalence queries ("does $g = f$?", f being the hidden function). The problem in the case of scientific discovery is that if experimentation results can be analyzed and interpreted to estimate the answer of a membership query, it is impossible to access an oracle able to answer an equivalence query ("is earth flat?"). By taking this point into account, and to improve the rule discovery process, we introduce a social interaction level by letting learner agents join a community respecting a multi-agent publication protocol to dispatch equivalence queries to other members of the community, as described in next section.

4.2 A Social Interaction Level to Cope with Equivalence Queries: Eleusis + Nobel

We designed the card game E+N to simulate a situation of collective problem solving implementing MQs. To simulate a real problem of scientific discovery, the oracle cannot be accessed to answer EQs. In fact, it is often hard and time consuming to determine the equivalence of two elaborated theories, which might not even use the same ontology, since each researcher has a personal way of describing the world and interpreting the experiment results.

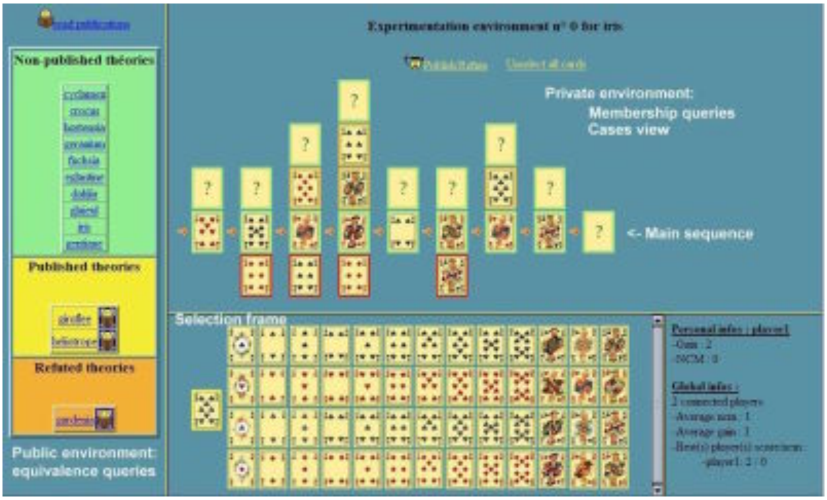


Fig. 4. Eleusis + Nobel Game display

Figure 4 shows a possible interface for E+N. This is the first we developed, as a web application, sufficient for the basic experimentations we made with human players only.

Private environment: The central frame displays the results of the experiments made by a player, by selecting cards in the bottom frame and placing them on the "? holes" to form a new sequence and submit it to the studied Nature's law. This way of displaying the results comes from Abbot's original game:

- A red surrounded card means it can not be placed after the previous one in the depicted main sequence.
- A green surrounded card means it forms a valid sequence with the previous card AND with the following one in the main sequence.
- When the card forms a valid sequence with the previous one in the main sequence BUT NOT with the following one, it is surrounded in orange.

Public environment: The frame on the left hand side displays a set of hidden rules, publicly accessible by an imaginary name so their meaning is hidden. A player can select the one he (or she) wants to study, and can switch between them whenever he wants to. It opens in the central frame a private experimentation context associated with the selected rule. When an *EQ* is formulated, by publishing a theory, the rule appears in the "published theories" cell, every player can read it, and the publisher scores *P* points. The theory is considered as correct until a player finds a counter example to refute it. The theory then moves to the "refuted theory" cell, the refuter wins R points, and the original publisher looses R points.

The P/R ratio can be set to modify the experimentation's conditions, and can also be different in the competitive scientific community and in a collaborative working group, as proposed in section 3.3.

The alternative to Equivalence queries is a publication, a broadcast of the player's theory to every other player belonging to the community. Each player can then compare this published theory with his personal current data, try to prove it's inconsistency, and refute it if a counter example is found. In fact, the theory is not broadcasted to everyone, but is added to a publicly accessible database storing every publication that is made on a hidden rule, and a notification is sent to players. This public database is a kind of collective memory, which efficiency as been shown in works on Case Based Reasoning as [16] or [17].

5 Results and Comments

We present in this section the results concerning the experimentation we made to validate our protocol. To reach this goal, we needed players, an experimentation which duration can be controlled, so the rules were defined to concern sequences of only two cards, in order to have a degree of difficulty suited to non assisted human players. The results showed that a human playing alone (i.e. using MQs alone) takes between 5 and 15 minutes to publish a theory concerning a rule implying only sequences of two cards. He usually considers his theory correct, and doesn't try to refute it. Moreover, the average number of published theories is between 10 and 20 (players stop before trying the 33 rules), and few of them are equivalent to the corresponding hidden rule.

An interesting alternative was to organise duels, between two players working on the same rule, until one of them admitted, without being sure, that the adverse theory was true. The players reached a consensus on a common Ontology.

To contrast with previous results, we made further experimentations involving multiple players, students coming from different scholar backgrounds. The average time for a publication is the same, and we observe a period of roughly half an hour during which players publish. Then they begin refuting each other, and theories are revised and republished. A community of ten to thirteen players takes between 1h 1/2 and 2h two reach a stable equilibrium of published theories (opposed to the theoretical length of 5h 1/2 for one-player games). The amount of correct theories is also much more superior. This empirically validate the need to use both Membership and Equivalence queries [1], and the use of a collective memory to share experience and points of view on a given problematic.

Some of these experimentations of the protocol failed because players made false refutations caused either by a misunderstanding of the ontology or of the published theory, or even of the notion of refutation. This shows the need, even for such small learning problems, to include in the publication a program simulating the user's theory and allowing to detect eventual contradictions within it (see section 2). A second notice is that the bias coming from the P/R ratio favoured the players who only refuted others' publications without publishing themselves. We need to define an other ratio taking into account Popper's idea that a falsifying experimentation shows a contradiction in a theory, but doesn't stand without a rival theory. The third notice is that this game is very efficient to teach the epistemological foundations of science theory.

Since the protocol is validated, we will now introduce the interaction between scientists and assisting machines, but this implies more time and more efforts from the players to learn how to work with adaptive problem solvers. This has to be worth it, so next experimentations will have to last longer, for example one month. These experimentations have excited some biologists who plan to help us designing another version of the game in which the hidden laws will be real scientific discoveries as described in [18], to simulate the (re)discovery of Nobel prizes... We will organize very soon an experimentation in which a team of human players will be opposed to a team of human players assisted by intelligent learning assistants to validate our approach. An obvious use of intelligent assistants is to let them test a sequence on various published theories, and to be the guaranteeing one of the user's published theory to answer MQs. The violation of the theory is located by the contradiction between the assistant's answer and what is really observed.

6 Conclusion

We presented an interaction model to assist scientists with adaptive problem solvers in their task of scientific discovery.

[1] formalized an interaction protocol for machine learning based on the use and combination of Membership queries and Equivalence queries, that enables a machine to learn a user's theory. In the context of scientific discovery, the user is fallible, and we emphasized that reasoning in a paraconsistent logic allows the solver to localize contradictions in the user's theory, which leads to a revision of the description model. Defeasible logic is useful to supervise the learning process and "$forget$" wrong theories. In our model, the user and the software assistant act in turns as the teacher or as the learner, and this interactive co-learning leads to a better understanding of the problem and to the creation of an adequate description model; being assisted by a learning machine trivializes some fairly easy problems.

Membership Queries can be simulated by designing experiments and interpreting their results, experimentation putting the hypothesis to test. To simulate the oracle to answer *Equivalence Queries*, we introduced a social cognition level to let multiple learners interact by answering each other's EQs. Stating that a group will solve a problem faster than an individual, we described a community of agents learning from each others, each having its own point of view and interpretation of events occurring in its environment, the learner refuting an EQ acting temporarily as the Teacher. Defining this interaction as a competition to optimize a score motivates the emission of queries.

This multi-agent discovery platform offers various industrial applications, especially as a tool for analysts trying to have a synthetic vision of a complex situation described by heterogeneous information sources, or for optimizing a production process involving complex systems.

References

1. Angluin, D., Krikis: Learning from different teachers. Machine Learning **51** (2003) 137–163
2. Nobrega, G.M.D., Cerri, Sallantin: A contradiction driven approach to theory information: Conceptual issues pragmatics in human learning, potentialities. Journal of the Brazilian Computer Society **9** (2003) 37–55
3. Angluin, D.: Queries and concept learning. Machine Learning **2** (1988) 319–342
4. Angluin, D.: Queries revisited. Theoretical Computer Science **313** (2004) 175–194
5. Popper, K.R.: Conjectures and Refutations: The Growth of Scientific Knowledge. Harper and Row (1963)
6. Cavaillès, J.: Sur la logique et la théorie de la science. Librairie Philosophique J. VRIN (1997)
7. Beziau, J.Y.: La logique paraconsistante. Logiques classiques et non classiques, essai sur les fondements de la logique (1997)
8. Nakamatsu, K., Kato, T., Suzuki, A.: Basic ideas of defeasible deontic traffic signal control based on a paraconsistent logic program evalpsn. Advances in Intelligent Systems and Robotics (2003)
9. Angluin, D.: Negative results for equivalence queries. Machine Learning **5** (1990) 121–150
10. L.Blum, Blum: Toward a mathematical theory of inductive inference. Inform. Control **28:2** (1975) 125–155
11. Gold, E.: language identification in the limit. Inform. Control **10** (1967) 447–474
12. Sallantin, J.: La découverte scientifique assistée par des agents rationnels. Revue des sciences et technologie de l'information (2003) 15–30
13. Liquière, M.: Structural machine learning with galois lattice and graphs. International Conference on Machine Learning - ICML (1998)
14. Chavalarias, D.: La thèse de Popper est-elle réfutable? Memoire de dea, CREA - CNRS/Ecole Polytechnique (1997)
15. Gardner, M.: Mathematical games. Scientific American (1959)
16. Cole, M., Engeström, Y.: A cultural historical approach to distributed cognition. Distributed Cognition (1993) 1–46
17. Garland, A., Alterman, R.: Multiagent learning through collective memory. Adaptation, Coevolution and Learning in Multiagent Systems: Papers from the 1996 AAAI Spring Symposium (1996) 33–38
18. Dunbar, K.: How scientists really reason: Scientific reasoning in real-world laboratories. Mechanisms of Insight (1995)

Cross-Language Mining for Acronyms and Their Completions from the Web

Udo Hahn[1], Philipp Daumke[2], Stefan Schulz[2], and Kornél Markó[2]

[1] Jena University Language and Information Engineerings (JULIE) Lab, Germany
http://www.coling.uni-jena.de
[2] Department of Medical Informatics, Freiburg University Hospital, Germany
http://www.imbi.uni-freiburg.de/medinf

Abstract. We propose a method that aligns biomedical acronyms and their long-form definitions across different languages. We use a freely available search and extraction tool by which abbreviations, together with their fully expanded forms, are massively mined from the Web. In a subsequent step, language-specific variants, synonyms, and translations of the extracted acronym definitions are normalized by referring to a language-independent, shared semantic interlingua.

1 Introduction

The understanding of acronyms and abbreviations in biomedical texts is crucial for various NLP applications, such as text mining [1], information extraction [2], or information retrieval systems [3]. This is witnessed, in particular, for protein and gene expressions from biomedical texts [4] (as well as the relations between them). Those expressions frequently consist of acronyms, but their definitions in the text might differ from the ones found, e.g., in external databases, such as ARGH, ACROMED, or SARAD [5] (cf. also [6] for an overview).

Multiple expansions for the same acronym, or multiple acronyms for the same definition, will lead to difficulties when one tries to match natural language expressions with a standardized vocabulary such as the UMLS or MESH [7]. In an information retrieval scenario, unresolved acronyms will possibly lead to a loss of precision: Does ”AD” refer to ”Alzheimer’s Disease” or to ”allergic dermatitis”? The problem of ambiguity becomes even harder when multilingual documents are encountered. This is likely to happen to Web search engines. In this case, the acronym ”AD” may have a German expansion (”atopische Dermatitis”), a Spanish one (”aurícula derecha”), or a Portuguese one (”agua destilada”), and possibly many more. Even worse, the German acronym equivalent to ”Alzheimer’s Disease” is ”AK” (”Alzheimer Krankheit”) or ”MA” (”Morbus Alzheimer”), while for Spanish the equivalent short-cut is ”EA”(”enfermedad de Alzheimer”).

Many research efforts have been spent on the automatic extraction of short-form/long-form (SF/LF) pairs (abbreviations and acronyms mapped to their expansions/completions) within a single language [8, 9, 10, 11, 5, 12, 13, 14]. Different ways of how abbreviations are actually used in written (medical) language were also studied [15], while little attention has been paid to how acronyms behave across languages.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 113–123, 2005.

This is a particular challenge for intelligent Web search engines and it is the focus of this paper.

2 Analysis of Terms into Subwords

We propose a method that automatically aligns acronyms and their definitions across different languages. It is based upon a dictionary the entries of which are *equivalence classes* of subwords, i.e., semantically minimal units [1]. From a linguistic perspective, subwords are often closer to formal Porter-style stems [2] rather than to lexicologically orthodox basic forms, e.g., of verbs or nouns or linguistically plausible stems. Hence, their merits have to be shown in experiments. These equivalence classes capture intralingual as well as interlingual synonymy. As equivalence classes abstract away from subtle particularities within and between languages and reference to them is realized via a language-independent concept system they form an *interlingua*.

Subwords are assembled in a multilingual lexicon and thesaurus, with the following considerations in mind:

- Subwords are listed, together with their attributes such as language (English, German, Portuguese, Spanish) or subword type (stem, prefix, suffix, invariant). Each subword is assigned one or more morpho-semantic class identifier(s), we call *MID*(s), representing the corresponding synonymy equivalence class.
- Intralingual synonyms and interlingual translation synonyms of subwords are assigned the same equivalence class (judged within the context of medicine only).
- Two types of meta relations can be asserted between synonymy classes:
  (i) a paradigmatic relation *has-meaning*, which relates one ambiguous class to its specific readings, as with:
  {*head*} ⇒ {*kopf,zephal,caput,cephal,cabec,cefal*} *OR* {*boss,leader,lider,chefe*}.
  (ii) a syntagmatic relation *expands-to*, which consists of predefined segmentations in case of utterly short subwords, such as:
  {*myalg*} ⇒ {*muscle,muskel,muscul*} ⊕ {*pain,schmerz,dor*}.[1]

We refrain from introducing additional hierarchical relations between MIDs because such links can be acquired from domain-specific vocabularies, e.g., the Medical Subject Headings [3] (cf. experimental evidence from Markó *et al.* [4]).

Figure 1 depicts how source documents (top-left) are converted into an interlingual representation by a three-step procedure. First, each input word is orthographically normalized in terms of lower case characters and according to language-specific rules for the transcription of diacritics (top-right). Next, words are segmented into sequences of subwords as found in the lexicon (bottom-right). Finally, each meaning-bearing subword is replaced by a language-independent semantic identifier, the corresponding MID, which unifies intralingual and interlingual (quasi-)synonyms, thus producing the interlingual output representation of the system (bottom-left). In Figure 1, bold-faced MIDs co-occur in both document fragments (after conversion into the interlingua format).

[1] '⊕' denotes the string concatenation operator.

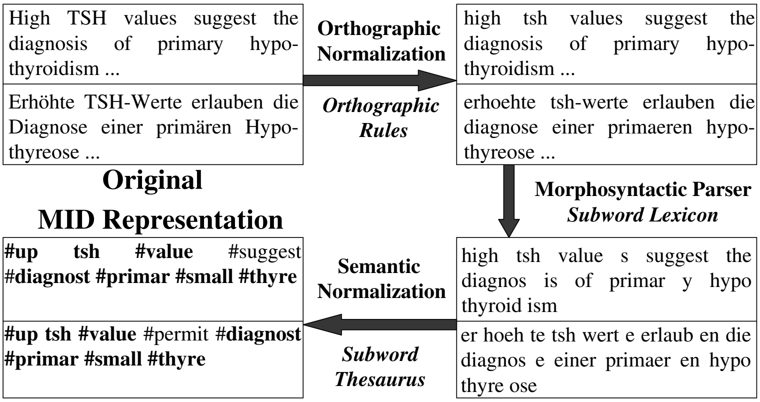

High TSH values suggest the diagnosis of primary hypothyroidism ...
Erhöhte TSH-Werte erlauben die Diagnose einer primären Hypothyreose ...
Original
Orthographic Normalization
Orthographic Rules
high tsh values suggest the diagnosis of primary hypothyroidism ...
erhoehte tsh-werte erlauben die diagnose einer primaeren hypothyreose ...
Morphosyntactic Parser
Subword Lexicon
high tsh value s suggest the diagnos is of primar y hypo thyroid ism
er hoeh te tsh wert e erlaub en die diagnos e einer primaer en hypo thyre ose
Semantic Normalization
Subword Thesaurus
MID Representation
#up tsh #value #suggest #diagnost #primar #small #thyre
#up tsh #value #permit #diagnost #primar #small #thyre

Fig. 1. Morpho-Semantic Indexing (MSI)

In the meantime, the entire subword lexicon (as of July 2005) contains 72,513 entries, with 22,067 for English,[2] 22,497 for German, 14,888 for Portuguese, and 13,061 for Spanish. All of these entries are related in the thesaurus by 20,990 equivalence classes. We also found a well-known logarithmic growth behavior as far as the increase of the number of subwords are concerned [1]. Under this observation, at least the English and German subword lexicons have already reached their saturation points.

Our project started from a bilingual German-English lexicon, while the Portuguese part was added in a later project phase (hence, its size still lags somewhat behind). All three lexicons and the common thesaurus structure were manually constructed, which took us about five person-years. While we simultaneously experimented with various subword granularities as well as weaker and stronger notions of synonymy, this manual approach was even heuristically justified. With a much more stable set of criteria for determining subwords emerging from these experiments, we recently switched from a manual to an automatic mode for lexicon acquisition. The Spanish sublexicon, unlike all other previously built sublexicons, was the first one generated solely by an automatic learning procedure which is specifically targeted at large-scale lexical acquisition. It makes initial use of cognate relations (roughly, string similarities) that can be observed for typologically related languages [5] and has recently been embedded into a bootstrapping methodology which induces new subwords that cannot be found by considering merely cognate-style string similarities. This extended acquisition mode makes heavy use of contextual co-occurrence patterns in comparable corpora [6].

In earlier experiments on cross-language information retrieval [1] and multilingual document classification [7], we showed the usefulness of representing medical documents on an interlingual layer. However, we were not able to properly account for acronyms, since they were completely missing in our lexicons. Therefore, we here

[2] Just for comparison, the size of WORDNET assembling the lexemes of general English in the 2.0 version is on the order of 152,000 entries (`http://wordnet.princeton.edu/man/wnstats.7WN`, last visited on May 13, 2005). Linguistically speaking, the entries are basic forms of verbs, nouns, adjectives and adverbs.

adapt previous work on automatic acronym detection to the needs of our interlingual representation approach.

3 Extracting Biomedical Acronyms and Completions

Our work reuses, without any modification, a simple and fast algorithm for the extraction of abbreviations and their completions from biomedical documents, which has been developed by Schwartz and Hearst [8].[3] The algorithm achieves 96% precision and 82% recall on a standardized test collection and, thus, performs at least as good as other existing approaches [9, 10, 11, 12, 13]. It deals with the extraction of acronyms and abbreviations together with their full forms (completions) in a two-step process. First, a list of candidate short-form/long-form (SF-LF) pairs is determined, which are then validated by taking additional selection criteria into account. In the following, we briefly describe the principles underlying both steps.

Extraction of possible SF-LF terms. Basically, SF-LF pairs are identified by their adjacency to parentheses. Two basic patterns, *LF (SF)* and *SF (LF)*, have to be distinguished. According to Schwartz and Hearst, a *short* form has the following characteristics: it contains between 2 and 10 characters, has a maximum of two words, at least one character is a letter, and its first character is alphanumeric. The *long* form must immediately appear before or after the corresponding short form and the maximum number of words is constrained by $min(|A| + 5, |A| * 2)$.[4] In practice, the *LF (SF)* pattern occurs more frequently. Therefore, only if a criterion for an *LF (SF)* pattern is not fulfilled (e.g., more than two words inside the parentheses), the complemenatry pattern, *SF (LF)*, is tried.

Selection of the correct SF-LF term. Next, rules are applied to identify the correct SF-LF pair from the list of candidates which were extracted in the first step. Most importantly, each character in the short form must match a character in the long form and characters of the short form must appear in the same linear order as in the long form. Furthermore, the first character of the SF has to be the same in the LF. Finally, all LFs are removed which are shorter than the corresponding SF, or which include the corresponding SF within one of their single words.

4 Experiments

The WWW is here taken as the authoritative textual resource where the largest and most up-to-date variety of acronyms and their associated completions can be found. Hence, for our experiments, we generated very large corpora directly from different, heterogeneous WWW sources, including MEDLINE. With more than 250m text tokens, the derived English corpus was much larger than those for the other languages involved (37m tokens for German, 14m for Portuguese, and 11m for Spanish, cf. Table 1). The

[3] The source code (in Java) is made available on the Web; see http://biotext.berkeley.edu/software.html.

[4] |A| is the number of characters in the corresponding SF.

Table 1. Corpus and Acronym Extraction Statistics

Language	Corpus Tokens	Proportion of Acronyms
English	250,258,039	1,253,311 (0.50%)
MSI-Covered		1,033,929 (82.5%)
German	37,105,363	29,967 (0.08%)
MSI-Covered		26,770 (89.3%)
Portuguese	13,904,790	8,532 (0.06%)
MSI-Covered		7,065 (82.8%)
Spanish	11,103,066	7,714 (0.07%)
MSI-Covered		4,723 (61,2%)

contribution of this paper lies in the cross-language linking of these data items by applying the MSI procedure outlined in Section 2.

Using the algorithm described above, we collected over 1.2m abbreviations together with their long forms for English, while we extracted some 30K pairs for German, 9K pairs for Portuguese and 8K pairs for Spanish (for exact numbers, cf. Table 1). In contradistinction to the other languages, the English corpus included a large number of expert-level MEDLINE abstracts. As a consequence, every 200th token in the collection was classified as an acronym. For the other languages (for which the corpora included a larger amount of consumer information), this ratio is much smaller (0.06 to 0.08 percent of the text tokens in the corpora).

After the acquisition of SF-LF pairs, the long forms were processed by the MSI procedure as described in Section 2. Upon prior manual inspection of document samples we observed that English long forms also tended to frequently occur in German,

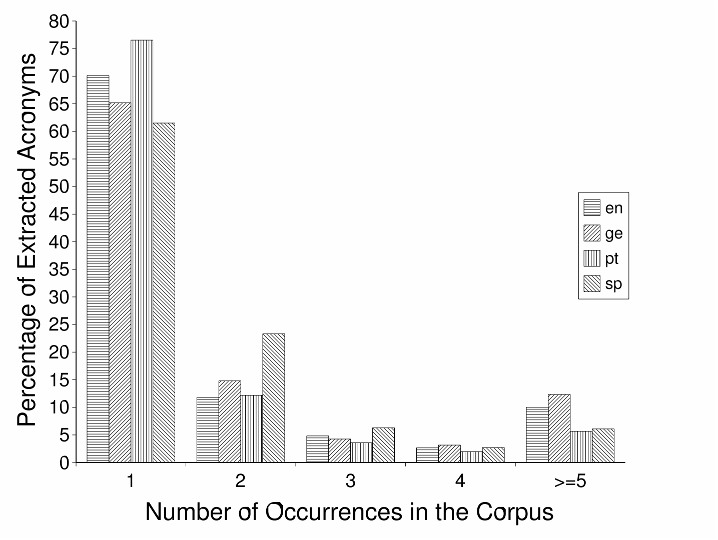


Fig. 2. Distribution of SF-LF Occurrences per Corpus

Table 2. Effects of Morpho-semantic Normalization in Terms of Unique SF-LF Pairs and Tokens per Type

Language	Surface		MSI	
	Unique	Ratio	Unique	Ratio
English	212,470	4.87	189,639	5.45
German	4,276	6.26	3,653	7.33
Portuguese	3,934	1.20	3,633	1.95
Spanish	2,037	2.32	1,911	2.47

Portuguese, and Spanish texts. Therefore, a decision had to be taken which lexicon to use for the MSI process. Our approach was to segment the long forms using *every* lexicon available (so no a priori decision was taken). Those language hypotheses were kept for which the underlying lexicon yielded *complete* lexical coverage with regard to the specific long form. If there were more than one remaining language hypothesis, the document language (if not English) was preferred over English.

This procedure led to over one million SF-LF pairs completely covered by the MSI procedure for English (83%), and approximately 27K pairs (89%) for German, 7K pairs (83%) for Portuguese, and 5K pairs (61%) for Spanish (cf. Table 1 for detailed numbers). In the following, we will only focus on this subset of extracted abbreviations. Figure 2 yields an impression of how frequent unique SF-LF pairs occur in the corpora considered, for each language condition. 61% to 76% of all acronyms extracted occur only once, 12% to 23% appear two times, whilst five or more occurrences are found for 6% to 12% of all SF-LF pairs.

As depicted in Table 2 (Column 2), 212,470 unique SF-LF pairs were generated for English, 4,276 for German, 3,934 for Portuguese, and 2,037 for Spanish. Column 3 of the table shows the average number of corpus occurrence for each unique SF-LP pair. After the MSI normalization of long forms, the number of unique SF-LF pairs decreases to 189,639 for English (3,653 for German, 3,633 for Portuguese and 1,911 for Spanish). Accordingly, the number of tokens per type increases, as depicted in the fifth column of Table 2. As an example, morpho-syntactic variants in long forms such as in *"CTC"*-*"computed tomographic colonography"* and *"CTC"*-*"computed tomography colonography"* are unified, an immediate effect of term normalization based on the interlingua (composed of equivalence classes of subwords).

4.1 Intra-Lingual Phenomena

Two basic ambiguity phenomena have to be considered when we discuss the results for a given language: First, one short form can have multiple long forms (SF ambiguity), and, second, one long form can have multiple short forms (LF ambiguity). An example for an SF ambiguity is given with *"ABM"* mapped to *"acute bacterial meningitis"* and to *"adult bone marrow"*. Table 3 shows the average numbers of different long forms for each short form, both for the baseline condition (lower-case surface form) and the MSI condition. For English, 82,501 unique short forms were extracted. The average number of long forms associated to unique SFs decreases from 2.56 to 2.30 for

Table 3. SF Ambiguity

Language	SFs	Average LF	
		Surface	MSI
English	82,501	2.56	2.30
German	2,954	1.45	1.24
Portuguese	2,517	1.56	1.44
Spanish	1,450	1.41	1.32

Table 4. LF Ambiguity

Language	Surface		MSI	
	LFs	Average SF	LFs	Average SF
English	184,639	1.15	154,693	1.23
German	4,187	1.02	3,515	1.04
Portuguese	3,798	1.04	3,395	1.07
Spanish	1,979	1.03	1,825	1.05

MSI, as expected. A similar tendency can also be observed for the other languages we considered.

The second phenomenon, one long form which comes with multiple different short forms, can also be observed in all languages involved in our experiments. For example, the noun phrase *"acid phosphatase"* has nine different abbreviations in the English corpus we processed (case insensitive): *"AcP"*, *"acPAse"* *"ACP-ase"*, *"Acph"*, *"ACPT"*, *"AP"*, *"APase"*, *"AphA"*, and *"APs"*. Table 4 depicts the numbers describing this phenomenon. For English, a total of 184,639 different long forms were extracted, arising from 212,470 different SF-LF pairs (cf. Table 2). Thus, each LF is associated with 1.15 SFs, on the average. For the MSI condition, fewer different long forms are encountered. Hence, the ratio slightly increases, for all languages.

4.2 Inter-Lingual Phenomena

4.2.1 Identical SF-LF Pairs

The first observation we made is that quite often SF-LF pairs are appear in other languages, such as *"WHO"* and its expansion *"World Health Organization"*, *"PCR"* and its completion *"polymerase chain reaction"*, or *"IL"* associated with *"interleukin"*. Summarizing (cf. Table 5, Column 2), we found 584 identical SF-LF for English-German, 181 for English-Portuguese, 192 for English-Spanish, 35 for German-Portuguese, 40 for German-Spanish, and 106 for Portuguese-Spanish (the latter sets also may contain some English SF-LF pairs).

4.2.2 Identical SF, Different LF

One way of identifying possible translations of long forms is to collect those long forms which are connected to a unique short form at the surface level. For example, if an English document contains *"WHO"*-*"World Health Organization"* and a German document contains *"WHO"*-*"Weltgesundheitsorganisation"*, the long forms can be regarded as possible translations of each other. For English-German, 100,915 of these pairs can be extracted, for English-Portuguese 151,037, for English-Spanish 109,568, for German-Portuguese 2,468, for German-Spanish 1,709, and for Portuguese-Spanish we counted 3,454 of these hypothesized translations (Table 5, Column 3). Of course, these sets also contain syntactic variants and a large number of false positives, since short forms are used differently across languages. Therefore, we switched our perspective to the interlingual layer of long form representations.

4.2.3 Identical SF, Translation of LF

In this condition, we examined those cases, in which short forms were identical and long forms were different at the surface level, but identical at the interlingual layer, by comparing SF-LF pairs extracted from the different source corpora. As a result, we obtained lists of bilingually aligned terms, such as English *"acute lymphatic leukemia"* linked to the German *"akute lymphatische Leukämie"* via the common short term *"ALL"*. As an example, 2,479 translations were generated for English-German using this heuristics (cf. Table 5 for additional data covering the remaining language pairs, as well).

Table 5. Statistics on Cross-Lingual Acronym Extraction: Results for Identical (I), Different (D) and Translations (T) of Short Forms (SF) and Long Forms (LF)

	Surface		MSI	
Language Pair	I(SF) I(LF)	I(SF) D(LF)	I(SF) T(LF)	D(SF) T(LF)
EN-GE	584	100,915	2,479	3,212
EN-PT	181	151,037	665	3,982
EN-SP	192	109,568	573	2,136
GE-PT	35	2,468	81	328
GE-SP	40	1,709	110	290
PT-SP	106	3,454	250	207
Total	1,138	369,151	4,158	10,155

4.2.4 Different SF, Translation of LF

In this scenario, we examined those cases, for which the long forms were identical or translations of each other (i.e., identical at the interlingua layer), but with different short forms. This captures interesting constellations such as English *"AIDS"* (*"acquired immune deficiency syndrome"*) aligned to Spanish or Portuguese *"SIDA"* (*"síndrome de inmunodeficiencia adquirida"*). We collected 207 of these translations for Portuguese-Spanish, and up to 3,212 for English-German (cf. Table 5, Column 5, for additional data covering the remaining language pairs, as well).

5 Lexicon Integration

In order to enhance the existing lexicons with acronyms automatically, the quality of the derived associations of short forms to long forms had to be ensured. To the best of our knowledge, we know of no multilingual acronym repository in the biomedical field which might serve as a suitable gold standard. With 96% precision, as measured by Schwartz and Hearst [8] on a standardized test set, we expect, however, about 8,500 false positives in the set of unique SF-LF pairs, only considering English (cf. Table 2). Furthermore, since our work focuses on cross-language information retrieval [1] and multilingual text classification [7], we are interested in the cross-lingual mapping of lexical entries. Both challenges are met by a simple heuristics, based upon the idea that "two languages are more informative than one" [14]. Hence, we incorporated those extracted SF-LF pairs in our subword lexicons, for which the long form is a translation of another, at least one, long form in a different language (after mapping on the interlingua layer). Thus, we collected those pairs for which the number of occurrences are depicted in Column 4 and 5 in Table 5. As a result, we obtained an intersection of 4,931 English SF-LF forms, and, correspondingly, 1,149 for German, 1,077 for Portuguese, and 647 for Spanish (a total of 7,804). For the monolingual mapping of short forms to long forms, we decided to additionally collect those language-specific SF-LF pairs, which occur at least 2 times on the layer of the interlingua (cf. Table 2, right). As Table 6 reveals, the lexicon size for the specific languages increased from initially 72,513 entries to 138,343 lexical items (61,081 new entries for English, 2,055 for German, 1,585 for Portuguese, and 1,109 for Spanish). Hence, our approach can truly be considered as a cross-language mining methodology for boosting lexicon growth through the incorporation of acronyms and abbreviations, as well as their associated completions.

Table 6. Enhancement of the Size of the Subword Lexicon

Language	Initial Size	New Acronyms
English	22,067	61,081
German	22,497	2,055
Portuguese	14,888	1,585
Spanish	13,061	1,109
Sum	72,513	65,830
Total	138,343	

6 Related Work

Several different techniques for the automatic extraction of abbreviations and their definitions from biomedical text (particularly from MEDLINE abstracts) have been developed up until now. Schwartz and Hearst [8] offer a simple and fast algorithm for the extraction of abbreviations and their completions from biomedical documents, to which we completely adhere in our approach. The algorithm achieves 96% precision and 82% recall on a standardized test collection and, thus, performs at least as good as other existing approaches [9, 10, 11, 12, 13].

Comprehensive databases with millions of entries are provided by different research groups [15, 9, 11, 12, 13]. They adopt similar sorts of heuristics such as identifying and processing parenthetical phrases within texts. Some of them rely on pattern matching only [16], some use stemming [9, 13], and/or apply term normalization routines to abbreviations and full forms [9, 11, 13] or employ statistical metrics [17]. In addition, Pustejovsky *et al.* [9] even incorporate a shallow parsing approach. A general overview of four large databases and their algorithms can be found in [18].

Our approach for the multilingual alignment of acronyms and their definitions is tied up to the research from these precursors. Unlike most previous research, however, we heavily exploit the WWW for gathering evidence for the linkage between abbreviations and their expanded forms. Furthermore, by mapping extracted long forms onto an interlingual representation layer, an approach which has not been considered so far, acronyms and their definitions are made comparable across different languages with a high coverage. The interlingua layer also serves as a conceptual filter to eliminate false friends (incorrectly linking short and long forms), which are likely to occur in a multilingual Web environment.

7 Conclusions

We introduced a method for aligning biomedical short forms (acronyms, abbreviations) and their associated long forms (completions) across four different languages. A total of 65,830 new lexicon entries were added to an already existing multilingual subword lexicon, boosting its original size by more than 90% of new lexical material.

Acknowledgements. This work was partly supported by Deutsche Forschungsgemeinschaft (DFG), grant KL 640/5-1, and the European Network of Excellence ”Semantic Mining” (NoE 507505).

References

[1] Schulz, S., Hahn, U.: Morpheme-based, cross-lingual indexing for medical document retrieval. International Journal of Medical Informatics **59** (2000) 87–99

[2] Porter, M.F.: An algorithm for suffix stripping. Program **14** (1980) 130–137

[3] MESH: Medical Subject Headings. Bethesda, MD: National Library of Medicine (2004)

[4] Markó, K., Hahn, U., Schulz, S., Daumke, P., Nohama, P.: Interlingual indexing across different languages. In: RIAO 2004 – Conference Proceedings: Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval, Avignon, France, 26-28 April 2004. Paris: Centre de Hautes Etudes Internationales d’Informatique Documentaire (CID) (2004) 82–99

[5] Schulz, S., Markó, K., Sbrissia, E., Nohama, P., Hahn, U.: Cognate mapping: A heuristic strategy for the semi-supervised acquisition of a Spanish lexicon from a Portuguese seed lexicon. In: COLING Geneva 2004 – Proceedings of the 20th International Conference on Computational Linguistics. Volume 2., Geneva, Switzerland, August 23-27, 2004. Association for Computational Linguistics (2004) 813–819

[6] Markó, K., Schulz, S., Medelyan, A., Hahn, U.: Bootstrapping dictionaries for cross-language information retrieval. In: SIGIR 2005 – Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005. New York, NY: ACM (2005)

[7] Markó, K., Daumke, P., Schulz, S., Hahn, U.: Cross-language MESH indexing using morpho-semantic normalization. In Musen, M.A., ed.: AMIA'03 – Proceedings of the 2003 Annual Symposium of the American Medical Informatics Association. Biomedical and Health Informatics: From Foundations to Applications, Washington, D.C., November 8-12, 2003. Philadelphia, PA: Hanley & Belfus (2003) 425–429

[8] Schwartz, A.S., Hearst, M.A.: A simple algorithm for identifying abbreviation definitions in biomedical text. In Altman, R.B., Dunker, A.K., Hunter, L., Jung, T.A., Klein, T.E., eds.: PSB 2003 – Proceedings of the Pacific Symposium on Biocomputing 2003, Kauai, Hawaii, USA, January 3-7, 2003. Singapore: World Scientific Publishing (2003) 451–462

[9] Pustejovsky, J., Castaño, J., Cochran, B., Kotecki, M., Morrell, M.: Automatic extraction of acronym-meaning pairs from MEDLINE databases. In Patel, V.L., Rogers, R., Haux, R., eds.: MEDINFO 2001 – Proceedings of the 10th World Congress on Medical Informatics. Vol. 1. Number 84 in Studies in Health Technology and Informatics, London, U. K., September 2001. Amsterdam: IOS Press (2001) 371–375

[10] Yu, H., Hripcsak, G., Friedman, C.: Mapping abbreviations to full forms in biomedical articles. Journal of the American Medical Informatics Association **9** (2002) 262–272

[11] Chang, J.T., Schütze, H., Altman, R.B.: Creating an online dictionary of abbreviations from MEDLINE. Journal of the American Medical Informatics Association **9** (2002) 612–620

[12] Wren, J.D., Garner, H.R.: Heuristics for identification of acronym-definition patterns within text: Towards an automated construction of comprehensive acronym-definition dictionaries. Methods of Information in Medicine **41** (2002) 426–434

[13] Adar, E.: SARAD: A simple and robust abbreviation dictionary. Bioinformatics **20** (2004) 527–533

[14] Dagan, I., Itai, A., Schwall, U.: Two languages are more informative than one. In: Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, CA, USA, 18-21 June 1991. Association for Computational Linguistics (1991) 130–137

[15] Rimer, M., O'Connell, M.: BIOABACUS: A database of abbreviations and acronyms in biotechnology and computer science. Bioinformatics **14** (1998) 888–889

[16] Taghva, K., Gilbreth, J.: Recognizing acronyms and their definitions. International Journal on Document Analysis and Recognition **1** (1999) 191–198

[17] Nenadić, G., Spasic, I., Ananiadou, S.: Automatic acronym acquisition and term variation management within domain-specific texts. In Rodriguez, M., Paz Suarez Araujo, C., eds.: LREC 2002 – Proceedings of the 3rd International Conference on Language Resources and Evaluation. Vol. 6, Las Palmas de Gran Canaria, Spain, 29-31 May, 2002. Paris: European Language Resources Association (ELRA) (2002) 2155–2162

[18] Wren, J.D., Chang, J.T., Pustejovsky, J., Adar, E., Garner, H.R., Altman, R.B.: Biomedical term mapping databases. Nucleic Acids Research **33** (2005) D289–293

Mining Frequent δ-Free Patterns in Large Databases

Céline Hébert and Bruno Crémilleux

GREYC, CNRS - UMR 6072, Université de Caen,
Campus Côte de Nacre
F-14032 Caen Cédex France
{Forename.Surname}@info.unicaen.fr

Abstract. Mining patterns under constraints in large data (also called fat data) is an important task to benefit from the multiple uses of the patterns embedded in these data sets. It is a difficult task due to the exponential growth of the search space according to the number of attributes. From such contexts, closed patterns can be extracted by using the properties of the Galois connections. But, from the best of our knowledge, there is no approach to extract interesting patterns like δ-free patterns which are on the core of a lot of relevant rules. In this paper, we propose a new method based on an efficient way to compute the extension of a pattern and a pruning criterion to mine frequent δ-free patterns in large databases. We give an algorithm (FTminer) for the practical use of this method. We show the efficiency of this approach by means of experiments on benchmarks and on gene expression data.

Keywords: Large databases, δ-free patterns, extensions, rules, condensed representations.

1 Introduction

Large data are data sets characterized by a large number of columns (i.e., attributes) and few rows (i.e., transactions). Data mining algorithms extracting patterns have difficulty in running on this kind of data because the search space grows exponentially according to the number of rows and it becomes huge. Known algorithms such as Apriori [1] or the recent algorithms that compute the so-called condensed representations can fail in mining frequent or constrained patterns in large data [17]. This is an important challenge because these geometrical dimensions are often encountered in a lot of domains (e.g., bioinformatics, quality control, text mining). For instance, in gene expression data, the matrices to explore gather the expression of tens of thousands of genes in few biological situations (we will see in Section 5 an example of such a matrix with 27,679 gene expressions and 90 biological situations). In quality control, the number of steps and parameters during the mass production is very numerous.

Extracting the complete collection of patterns under various kind of constraints in such data is a promising direction research . The completeness means that every pattern which satisfies the defined constraints has to be returned

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 124–136, 2005.

(e.g., every frequent pattern, every closed pattern). This is important to capture all the information embedded in the data. For instance, in biological data, frequent patterns are on the basis of synexpression groups (i.e., co-regulated sets of genes assumed to take part in a common function within the cell). Thanks to the properties of Galois connections and the transposition of data, a technique has been proposed in the particular case of closed patterns [17]. Unfortunately, we will see Section 2.2 that this approach of transposition is impracticable with the δ-free patterns.

In this paper, we focus on the search of free (or key) patterns [4, 14] and δ-free patterns [6]. The latter are a generalization of free patterns. Let us recall that free patterns are made of attributes without relations among them. They reveal the sound relationships between the data. With regard to the constraint of frequency, they are the minimal patterns of the classes of equivalence. As the property of freeness (and δ-freeness) is anti-monotonous, free and δ-free patterns can be efficiently extracted even in correlated data [6]. These patterns make an efficient condensed representation of all frequent patterns and their uses are highly interesting. They enable to build rules with a bounded number of exceptions [5], non redundant rules [18], their capacity to indicate the minimal part of attributes highlighting a phenomenon is precious in classes characterization and classification [3, 7]. δ-free patterns combine the exhaustiveness of the relations within the database and the simplicity which is required to build rules (and especially classification rules) without over-fitting. There is a need of classes characterization and classification techniques in large data, for instance, to predict a cancer diagnosis according to individual gene expression profiles or, in the area of the quality control, to detect an equipment which is badly tuned in a silicon plate production chain.

We propose in this paper a method to mine frequent and δ-free patterns from large data without transposing the data set. The key idea is to use the extension of a pattern to check these constraints, because the extension has few objects in large databases. We show a new property to compute the extension of a pattern and a new pruning criterion. Their simultaneous use is on the core of the FTminer algorithm that we propose to extract the frequent and δ-free patterns from data. Then we show the efficiency of FTminer by means of experiments on several benchmarks and a gene expression database.

The organization of this paper is as follows. In Section 2, we recall useful definitions and we discuss related work on δ-free patterns mining. Section 3 presents our approach and new properties on extensions and pruning. The algorithm FTminer is given in Section 4. We end this paper by some experimental results in Section 5.

2 Context and Definitions

2.1 Notations and Definitions

Basic notations. Let us recall some definitions and notations useful for the rest of this paper. We define a database r as a relation $\mathcal{R}$ between the set $\mathcal{A}$ of *attributes* (or *items*) and the set $\mathcal{O}$ of *objects* (or *transactions*): for $a \in \mathcal{A}, o \in \mathcal{O}$, $a\mathcal{R}o$ if and only if the object o contains the attribute a. r can also be viewed as a boolean matrix. In this case, we say that $a\mathcal{R}o$ if and only if $(a, o) = 1$ in the matrix. Notice that $o \in \mathcal{O}$ is also a set of attributes. An *attribute pattern* or *itemset* is a subset of $\mathcal{A}$. Similarly, an *object pattern* is a subset of $\mathcal{O}$. We say that an attribute pattern X is supported by an object o if o contains X. A *specialization* relation is defined on the attributes patterns (resp. objects patterns): a pattern X_1 is more specific than X_2 if X_2 is a subset of X_1. Thanks to this relation, the attribute patterns can be represented in a *lattice*. We give an example of *transactional database* in Table 1.

Table 1. An example of transactional database

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
o_1	1	0	1	0	1	0	1	0
o_2	0	1	1	0	1	0	1	0
o_3	1	0	1	0	1	0	0	1
o_4	1	0	0	1	0	1	0	1
o_5	0	1	1	0	0	1	0	1

γ-frequency and δ-freeness. An attribute pattern X is *γ-frequent* if it is supported by at least γ objects in r, γ being a given threshold. In Table 1, the frequency (noted $\mathcal{F}$) of the attribute pattern a_3a_5 is 3 (i.e., $\mathcal{F}(a_3a_5) = 3$) and a_3a_5 is said 3-frequent. X is a *δ-free* pattern if there is no association rule between two of its proper subsets with less than δ exceptions (i.e., there is no rule $X_1 \Rightarrow X_2$ with $\mathcal{F}(X_1 \cup X_2) + \delta \geq \mathcal{F}(X_1)$ and $X_1 \cup X_2 = X$ and $X_1 \cap X_2 = \emptyset$). To facilitate the understanding of the next sections, we will use the following equivalent definition of the δ-freeness [6]: X is a *δ-free* pattern if for each $X_1 \subset X$, $\mathcal{F}(X) + \delta < \mathcal{F}(X_1)$. In Table 1, a_5a_8 is 1-free since $\mathcal{F}(a_5a_8) = 1$ and one have $\mathcal{F}(a_5) = \mathcal{F}(a_8) = 3 > \mathcal{F}(a_5a_8) + \delta$. When $\delta = 0$, X is called a 0-free set or a *free* set.

Extension and Intension. We recall the definition of the extension of an attribute pattern. Let X be an attribute pattern, O an object pattern. The *extension* $g(X)$ is the maximal set of the objects containing X. The *intension* $f(O)$ is the maximal set of attributes appearing in every object of O. $h = f \circ g$ and $h' = g \circ f$ are the closure operators of the Galois connection. An attribute (resp. object) pattern X (resp. O) is *closed* if $h(X) = X$ (resp. $h'(O) = O$).

2.2 Related Work

The minimal frequency constraint is the most usual constraint in data mining. It is on the core of well-known algorithms like APRIORI [2] which extracts all the γ-frequent patterns by scanning the database at each level. This levelwise algorithm is generalized to anti-monotonous constraints according to the specialization of attributes [12]. If this technique is efficient in sparse data, it fails in correlated data [5]. By computing the frequency of only a few patterns (the minimal and the maximal patterns of the classes of equivalence), condensed representations based on free (or key) patterns [4, 14] and on closed patterns [5, 13, 15, 19] improve this approach on correlated data. These condensed representations are called *exact* because the exact frequency of each pattern can be inferred. If a bounded number of errors on the frequency of patterns is accepted, the condensed representation of δ-free patterns is more concise and can be mined more efficiently. Let us note that the δ-freeness is an anti-monotonous constraint and the higher δ, the more the efficiency of the pruning increases. It is important to be able to extract these patterns because they enable multiple uses on data mining techniques [3, 11, 18] (e.g., rules with minimal body, characterization of classes, classification).

Unfortunately, if the number of attributes is very large (i.e., large data), even the algorithms on condensed representations based on closed and δ-free patterns fail (except if the frequency threshold is very high, which is not sensible in real applications). In the specific case of the closed patterns, a technique relying on the properties of Galois connections and the transposition of data has been proposed [17]. Unfortunately, there is no straightforward generalization of this approach for a lot of constraints. By using this technique, the extracted patterns are object patterns and no longer attribute patterns and it is necessary to define the transposed form of the constraint. It is easy for closed patterns (thanks to the Galois connections), but not for a lot of constraints and especially for the δ-freeness [10]. In this case, each equivalence class contains at least one constrained pattern and one has to consider each attribute pattern of the lattice [10].

Notice that another method to extract free patterns is presented in [16]. It uses generalized properties on antimatroid spaces. An antimatroid space corresponds to the particular case of a lattice where each equivalence class of frequency contains one unique minimal generator. It is unlikely that happens in real data sets but this method has been extended to spaces that are not antimatroids in [9]. In this last case, the free patterns can be extracted from the closed ones by using minimal transversals of hypergraphs, but the complexity of the technique remains an open issue [8] and this approach cannot be used in practice.

3 Computing Frequency and δ-freeness in Large Data

This section presents our approach to mine frequent δ-free patterns in large data. We start by giving the main ideas, then we specify the technical key points: the link between extensions and the minimal frequency and δ-freeness constraints, our technique to compute the extensions and a new criterion based on the conjunction of the minimal frequency and δ-freeness constraints.

3.1 Main Ideas of Our Approach

The computation of the closures of patterns is often a bottleneck for algorithms mining frequent and δ-free patterns. Unfortunately, in the case of large databases, the closures contain a lot of attributes and their storage requires a large amount of memory. That is why this approach often fails. But, with large data, there are only few objects which satisfy a set of attributes. Our idea is to check the δ-freeness constraint by using the corresponding patterns of objects: the extensions gather small object patterns easier to store.

Let us note that γ-frequency and δ-freeness are anti-monotonous constraints. We benefit from the pruning properties coming from such constraints. Moreover, we define and exploit a new pruning criterion stemmed from the conjunction of the γ-frequency and δ-freeness. This criterion is checked by using the extensions of patterns. Finally, we think that the success of this approach lies on the combination of these two points: mining γ-frequent and δ-free patterns by using the extensions and the use of this new pruning criterion.

3.2 Extension as a Frequency

Property 1 indicates the relation between the extension and the frequency of an attribute pattern.

Property 1. *The frequency of an attribute pattern X is equal to the cardinal of its extension $|g(X)|$.*

It is clear that Property 1 is well known but its use is interesting because it enables to rewrite the definitions of the minimal frequency and δ-freeness constraints with extension:

Definition 1. *An attribute pattern X is γ-frequent if $|g(X)| \geq \gamma$.*

Definition 2. *An attribute pattern X is δ-free if for all $X_1 \subset X$, $|g(X)| + \delta < |g(X_1)|$.*

In the example in Table 1, the extension of the attribute pattern a_1a_3 is equal to o_1o_3 and its frequency is 2 as indicated by Property 1. a_1a_3 is 2-frequent. To illustrate Definition 2, let us have a look at the patterns a_1a_3 and a_1a_4. a_1a_3 is 0-free because $|g(a_1)| = 3 > |g(a_1a_3)| + \delta = 2$ and $|g(a_3)| = 4 > |g(a_1a_3)| + \delta$. Nevertheless, a_1a_4 is not 0-free since $|g(a_4)| = 1 = |g(a_1a_4)| + \delta$.

An immediate and important consequence of Definitions 1 and 2 is that we are now able to establish the frequency and the δ-freeness of any pattern only with its extension. The next section explains how to compute efficiently the extensions.

3.3 A New Property to Compute Extension

The Property 2 allows to compute the extension of a pattern X from the extension of two of its subsets provided that their union is equal to X. So, from the extensions of the patterns at the level k, we are able to determine the extensions of the patterns at the level $k + 1$.

Property 2. *Let X_1 and X_2 be two patterns, the extension of $X_1 \cup X_2$ is equal to $g(X_1) \cap g(X_2)$.*

Proof. $\subseteq$ We have $X_1 \subseteq X_1 \cup X_2$ and $X_2 \subseteq X_1 \cup X_2$. As g is a decreasing function, we obtain that $g(X_1 \cup X_2) \subseteq g(X_1)$ and $g(X_1 \cup X_2) \subseteq g(X_2)$ so we have immediately $g(X_1 \cup X_2) \subseteq g(X_1) \cap g(X_2)$.
$\supseteq$ Let us consider o an object of $g(X_1) \cap g(X_2)$. By definition, o contains the patterns of attributes X_1 and X_2. As a consequence, we deduce that o contains $X_1 \cup X_2$. So o belongs to $g(X_1 \cup X_2)$.

For instance, in the example in Table 1: $g(a_1a_8) = o_3o_4$, $g(a_3a_5) = o_1o_2o_3$ and $g(a_1a_3a_5a_8) = o_3 = g(a_1a_8) \cap g(a_3a_5)$.

Several advantages stem from this property for mining patterns in large data. Firstly, as already said, the extensions are short patterns, easy to store and their intersections are computed in a short time. Secondly, to get the extension of a pattern X, we only have to compute the intersection of the extensions of two subsets of X (and not of all its subsets). Thirdly, the database is only scanned *once* (for patterns of length 1, i.e., items). On the contrary of the running of an usual levelwise algorithm, this avoids storing for each level of the search space all the candidate patterns.

We will see in Section 4 that it is sufficient to use Property 2 on patterns having the same length and a common prefix to mine frequent δ-free patterns in large data.

3.4 A New Pruning Criterion

This section presents a new pruning criterion for mining frequent δ-free patterns. First, let us note that, as both the frequency and the δ-freeness are anti-monotonous constraints, we naturally use the efficient pruning properties linked to such constraints [12]. Nevertheless, we go further and we highlight a new pruning criterion (Criterion 1) which comes from Theorem 1. This new pruning criterion is based on the common use of the minimal frequency and the δ-freeness properties.

Theorem 1. *Let X be a pattern. If X is a γ-frequent and δ-free pattern then for all $X_1 \subset X$, $|g(X_1)|$ is greater than $\gamma + \delta$.*

Proof. Theorem 1 is an immediate consequence of Definitions 1 and 2. X is a γ-frequent and δ-free pattern. Definitions 1 and 2 imply that for all $X_1 \subset X$, $\gamma + \delta \leq |g(X)| + \delta < |g(X_1)|$.

Pruning Criterion 1. *Let X be a pattern such that $|g(X)| \leq \gamma + \delta$, there is no superset of X being a γ-frequent δ-free pattern. So a levelwise algorithm can prune the search space from X.*

Criterion 1 is obtained by the contrapositive of Theorem 1. Let us examine the pattern a_5a_7 in the example in Table 1. a_5 and a_7 are 1-frequent and 1-free. They cannot be pruned by using classical pruning properties of anti-monotonous constraints and a_5a_7 is generated. Nevertheless, by using Criterion 1, a_5a_7 is not a candidate because $|g(a_7)| = 2 = \gamma + \delta$. The explanation of this pruning is the following. To be 1-frequent, $|g(a_5a_7)|$ should be greater than or equal to 1. But, to be 1-free, $|g(a_5a_7)|$ should be smaller than $|g(a_7)| - 1 = 1$. So, the minimal frequency is in contradiction with the δ-freeness and a_5a_7 cannot satisfy simultanously these constraints.

4 FTminer

This section presents FTminer (FT for Free faT[1] databases Miner), an algorithm based on our approach given in Section 3. FTminer extracts all the γ-frequent δ-free patterns from a database r. It follows the outline of levelwise algorithms. Let us recall that its originality is that there is no generation phase for all candidates which is very useful for large data. The database is only scanned once (for items) and, thanks to the use of extension, generation and verification are simultaneous. The process is also speeded up by the pruning Criterion 1.

```
FTminer ( database $r$, threshold $\gamma$, number of exceptions $\delta$ )

$\mathcal{F}ree_1 := \{a \in \mathcal{A} \mid |\mathcal{O}| - \delta > |g(a)| \geq \gamma\}$
$\mathcal{G}en_1 := \{a \in \mathcal{F}ree_1 \mid |g(a)| > \gamma + \delta)\}$
$k := 1$
while $\mathcal{G}en_k \neq \emptyset$ do
    for each $(Y \cup \{A\}, Y \cup \{B\}) \in \mathcal{G}en_k \times \mathcal{G}en_k$ do
        // generation of one candidate of length $k+1$
        $X := Y \cup \{A\} \cup \{B\}$
        $g(X) := g(Y \cup \{A\}) \cap g(Y \cup \{B\})$
        // $\gamma$-frequency
        if $|g(X)| \geq \gamma$ then
            // $\delta$-freeness
            $i := 1$
            while $i \leq k+1$ and $X \backslash \{x_i\} \in \mathcal{G}en_k$ and
            $|g(X)| + \delta < |g(X \backslash \{x_i\})|$ do
                $i := i + 1$
            od
            if $i = k + 2$ then
                $\mathcal{F}ree_{k+1} := \mathcal{F}ree_{k+1} \cup \{X\}$
                if $|g(X)| > \gamma + \delta$ then
                    $\mathcal{G}en_{k+1} := \mathcal{G}en_{k+1} \cup \{X\}$
    od
    $k := k + 1$
od
return $\bigcup_{i=1}^{k-1} \mathcal{F}ree_i$
```

[1] The word "fat" is also used to refer to large data sets as indicated for instance by D. Hand during his invited talk at PKDD'04.

Let $\mathcal{F}ree_k$ be the set of the free patterns at the level k whose frequency is greater than or equal to γ and $\mathcal{G}en_k$ be the set of generators at the level k i.e., the patterns in $\mathcal{F}ree_k$ with a frequency greater than $\gamma + \delta$.

The first step is the initialization of $\mathcal{F}ree_1$ and $\mathcal{G}en_1$. One scan on the database enables to compute the extension of the items and to determine whether they are γ-frequent and δ-free or not and $\mathcal{F}ree_1$ is obtained (Line 1). The initialization of $\mathcal{G}en_1$, using the pruning Criterion 1, stands in Line 2 and $\mathcal{G}en_1$ contains the γ-frequent δ-free patterns which have a frequency greater than $\gamma + \delta$.

The main loop begins in Line 4: it stops when there is no generators left at the considered level. For generating one candidate X at the level $k+1$ (Line 5), two patterns having a common prefix Y of length $k-1$ are joined (Line 6). The computation of the extension of X by intersecting the extensions of its generators is performed Line 7 using Property 2. In Line 8, Definition 1 is used to test whether the candidate X is γ-frequent thanks to its extension.

The loop begining at Line 10 considers every subset of X of length k. For each one (except for the two generators) the algorithm checks if it belongs to $\mathcal{G}en_k$ (i.e., if it is γ-frequent, δ-free and if its frequency is greater than $\gamma + \delta$) and if its frequency is greater than the frequency of the candidate plus δ. If X satisfies all the tests (Line 12), it is added in $\mathcal{F}ree_{k+1}$. Moreover, X is also a generator if its frequency is greater than $\gamma + \delta$ using Criterion 1.

Theorem 2 shows that FTminer is correct and complete.

Theorem 2. *The algorithm* FTminer *extracts all the γ-frequent and δ-free patterns from the database r.*

Proof (Correctness). Let us prove that a pattern X in $\mathcal{F}ree_k$ is a γ-frequent δ-free pattern. We test at Line 8 if $|g(X)| \geq \gamma$, what ensures that X is γ-frequent. At Line 10, we establish that X is δ-free using the condition $|g(X)| + \delta < |g(X\backslash\{x_i\})|$ (cf. Definition 2).

Proof (Completeness). The algorithm FTminer covers the whole attribute search space thanks to the principle of the levelwise algorithms. The accuracy of the used pruning criteria (properties of anti-monotonous constraints and Criterion 1) entails the completeness of FTminer.

5 Experiments

The aim of the experiments is to show the run-time benefit brought by FTminer and emphasizes that FTminer is able to mine frequent δ-free patterns in situations where prototypes (even taking benefit from condensed representations) fail. In Section 5.1 we compare on benchmarks FTminer to MVminer. The latter is a common prototype to extract condensed representations composed of δ-free patterns[2]. Let us note that it is equivalent to ACminer implemented by

[2] MVminer has been implemented by François Rioult (GREYC).

A. Bykowski (LIRIS) [5]. To the best of our knowledge, there exists no other prototype to mine frequent δ-free patterns. Section 5.2 widens the comparison to real gene expression data sets.

All the tests were performed on a 2.20 GHz Pentium IV processor with Linux operating system by using 3Go of RAM memory.

5.1 Results on Benchmarks

The benchmarks come from the UCI repository[3].

Benchmarks with a Lot of Attributes. In order to get large benchmarks, we transposed the CMC and ABALONE data sets. Thus, in the following, the used data sets have 30 rows and 1474 columns for CMC, 30 rows and 4178 columns for ABALONE. Figure 1 plots the comparison between FTminer and MVminer on run-times during the computation of frequent 0-free patterns according to the frequency threshold γ. γ ranges from 10 to 6 (37 to 20 percent) for CMC, 9 to 6 (30 to 20 percent) for ABALONE.

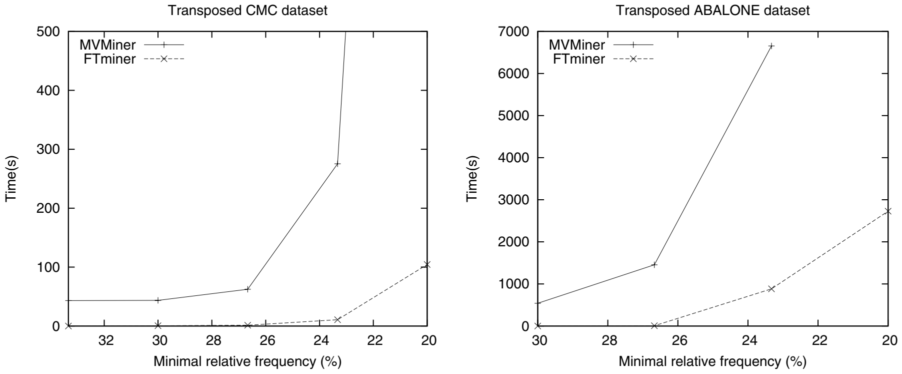


Fig. 1. Run-time performances according to the frequency on large data

As expected, the run-time increases when γ decreases. FTminer clearly outperforms MVminer (Figure 1). The latter fails when γ is equal to 5 (17%) for lack of memory while FTminer ends in 2420 s on CMC. MVminer also fails on ABALONE when γ is equal to 6 (20%).

Benchmarks with Usual Dimensions. Out of curiosity, we test FTminer on data with usual dimensions i.e. having much more rows than attributes. We used the benchmarks MUSHROOM and PUMSB (from UCI repository). MUSHROOM is a 8124 × 120 data and PUMSB a 49046 × 7118 data. Figure 2 indicates that FTminer runs faster than MVminer even if there is an important number of

[3] http://www.ics.uci.edu/~mlearn/MLSummary.html

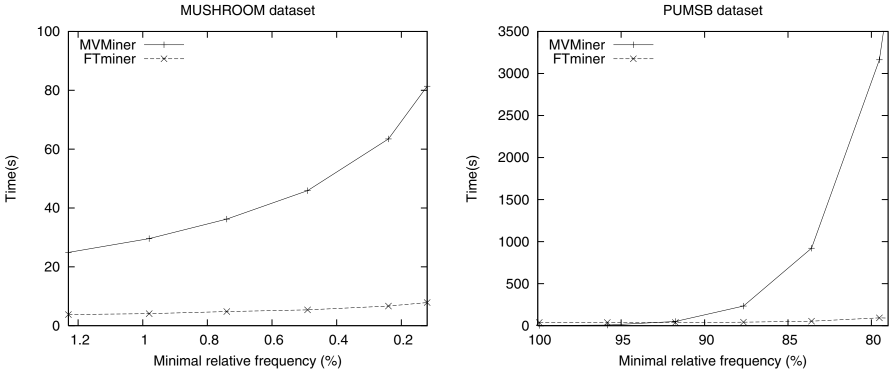


Fig. 2. Run-time performances according to the frequency on data having usual dimensions

objects in MUSHROOM and PUMSB. However, in one situation (PUMSB benchmark with a relative frequency threshold of 75.5%), FTminer was lacking of memory due to the size of the extensions while MVminer ends in 8829 seconds. This result was expected because the benefit of using the extension on large data (i.e., few patterns of objects) might not be reached on huge data with usual dimensions.

5.2 Results on Gene Expression Data Sets

We performed similar comparisons on a publicly available human Serial Analysis of Gene Expression (SAGE) data set[4] SAGE is an experimental technique designed to quantify gene expression. SAGE data provide expression values for given biological situations and given genes. These data sets are characterized by a large number of columns and few biological situations. For instance, the data set used for these experiments gathers 27,679 gene expressions for 90 biological situations.

Figure 3 (left) plots the run-times for mining the 3-free patterns with γ varying from 30 to 24 (33 to 27 percent). We used a logarithmically scaled ordinate axis. With a relative frequency threshold of 33.3%, FTminer spends 30 seconds whereas one day is needed for MVminer. With a threshold of 32%, FTminer spends 50 seconds and MVminer more than two days. Such results show the efficiency of FTminer on large data.

Another aim was to experimentally quantify the efficiency of the new pruning criterion (Criterion 1). Figure 3 (right) plots the run-times of the extractions with and without this pruning criterion according to the number of exceptions. The run-time benefit is important: for $\gamma = 27$ (corresponding to 30%) and $\delta = 5$, it spends 31 seconds to extract the frequent δ-free patterns using Criterion 1 and 527 seconds without. In average, the run-time is divided by 7 thanks to

[4] This data set comes from the CGMC laboratory (CNRS UMR 5534) and has been prepared by Olivier Gandrillon and Sylvain Blachon.

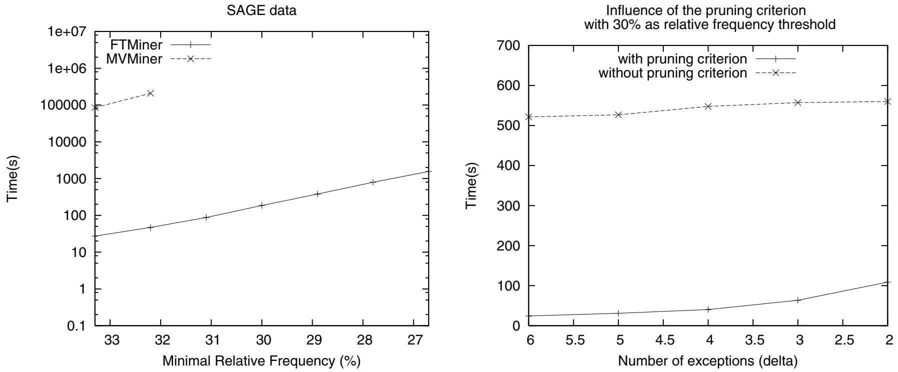


Fig. 3. Run-time performances on SAGE data

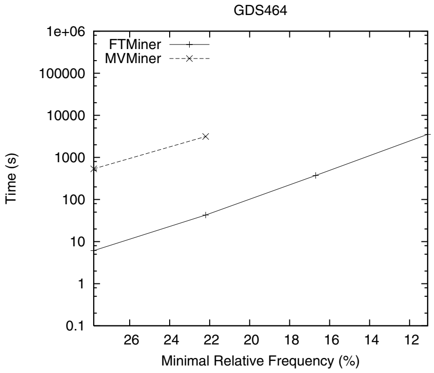


Fig. 4. Run-time performances on Dual-Channel data (GDS464)

Criterion 1. This can be explained by the large number of candidates additionally pruned thanks to this criterion: when $\gamma = 27$ and $\delta = 5$, this number is divided by 52, from 732,557,270 to 14,056,991.

Obviously this approach runs on other gene expression data sets. Out of curiosity, we ran our prototype on the gene expression data `GDS464` from the Gene Expression Omnibus repository[5] This data (collected in Dual-Channel experiments) gives the expression of 7085 genes in 90 biological situations. Figure 4 shows the run-times for mining the 2-free patterns according to γ (logarithmically scaled ordinate axis). The extraction becomes intractable with MVMINER when γ is less than 20%.

These experiments show that using both the extensions and the new pruning criterion enables to mine frequent δ-free patterns in large data whereas other approaches fail.

[5] Publicly available at URL `http://www.ncbi.nlm.nih.gov/projects/geo/gds/gds_browse.cgi?gds=464`

5.3 Discussion

These experiments prove the practical interest of the use of the extension in the case of large data sets. Nevertheless, Section 5.1 shows that the use of extension could also be efficient when mining data with usual dimensions (i.e., a large number of objects and few attributes). Furthermore, even in such data, the computational cost of the closures is more expensive than the one of the extension. It may be explained by the fact that the computing of a closure requires to intersect all the objects containing a given pattern in the data set. The computing of an extension is purely limited to the intersection of two objects as explained in Section 3.3.

6 Conclusion

Mining patterns in large data is a difficult task due to the large number of attributes. It is an important challenge because a lot of data sets have such geometrical dimensions and patterns like frequent δ-free are required by the owners of the data for several uses like classes characterization or classification. In this paper, we have proposed a new method based on a efficient way to compute the extension of a pattern and a pruning criterion to mine frequent δ-free patterns in large databases. A key point of the success of this approach is that the extensions in large data gather small object patterns easy to store. Experiments on benchmarks and a real gene expression data set show the practical use of this approach. Further work deals with the use of the extension to improve the extraction of patterns satisfying other constraints.

Acknowledgements. The authors thank the CGMC Laboratory (CNRS UMR 5534, Lyon) for providing the gene expression database. We thank François Rioult and Arnaud Soulet for fruitful discussions. This work has been partially funded by the ACI "masse de données" (MD 46, Bingo).

References

[1] R. Agrawal, H. Mannila, R. Srikant, and H. Toivonen. Fast discovery of associations rules. In *Advances in Knowledge Discovery and Data Mining*, 1996.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.

[3] R. J. Bayardo. The hows, whys, and whens of constraints in itemset and rule discovery. In *proceedings of the workshop on Inductive Databases and Constraint Based Mining*, 2005.

[4] J.-F. Boulicaut and A. Bykowski. Frequent closures as a concise representation for binary data mining. In *proceedings of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'00)*, volume 1805 of *Lecture notes in artificial intelligence*, pages 62–73, Kyoto, Japan, 2000. Springer-Verlag.

[5] J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queries by means of free-sets. In *The Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases*, Lyon, 2000.

[6] J. F. Boulicaut, A. Bykowski, and C. Rigotti. Free-sets: a condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery journal*, 7(1):5–22, 2003.

[7] B. Crémilleux and J.-F. Boulicaut. Simplest rules characterizing classes generated by delta-free sets. In *22nd Int. Conf. on Knowledge Based Systems and Applied Artificial Intelligence (ES'02)*, pages 33–46, Cambridge, UK, December 2002.

[8] 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.

[9] R. E. Jamison and J. L. Pfaltz. Closure spaces that are not uniquely generated. In *Ordinal and Symbolic Data Analysis*, Brussels, 2000.

[10] B. Jeudy and F. Rioult. Database transposition for constrained closed pattern mining. In *Third International Workshop on Knowledge Discovery in Inductive Databases*, 2004.

[11] H. Mannila and H. Toivonen. Multiple uses of frequent sets and condensed representations. In *proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, pages 189–194, Portland, Oregon, 1996.

[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] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. *Lecture Notes in Computer Science*, 1999.

[14] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Data Mining and Knowledge Discovery journal*, 24(1):25–46, 1999.

[15] J. Pei, J. Han, and R. Mao. CLOSET: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[16] J. L. Pfaltz and C. Taylor. Closed set mining of biological data. In *Workshop on Data Mining and Bioinformatics*, 2002.

[17] F. Rioult, J.-F. Boulicaut, B. Crémilleux, and J. Besson. Using transposition for pattern discovery from microarray data. In M. J. Zaki and C. C. Aggarwal, editors, *proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'03)*, pages 73–79, San Diego, CA, 2003.

[18] M. J. Zaki. Generating non-redundant association rules. In *proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD'00)*, pages 34–43, Boston, MA, 2000.

[19] M. J. Zaki and C. J. Hsiao. CHARM: an efficient algorithm for closed itemset mining. In *proceedings of the 2th SIAM international conference on Data Mining*, pages 33–43, Arlington, 2002.

An Experiment with Association Rules and Classification: Post-Bagging and Conviction*

Alípio M. Jorge and Paulo J. Azevedo

[1] LIACC, Faculdade de Economia, Universidade do Porto, Rua de Ceuta, 118,
4050-090, Porto, Portugal
amjorge@fep.up.pt
[2] Departamento de Informática, Universidade do Minho, Portugal
pja@di.uminho.pt

Abstract. In this paper we study a new technique we call post-bagging, which consists in resampling parts of a classification model rather then the data. We do this with a particular kind of model: large sets of classification association rules, and in combination with ordinary best rule and weighted voting approaches. We empirically evaluate the effects of the technique in terms of classification accuracy. We also discuss the predictive power of different metrics used for association rule mining, such as confidence, lift, conviction and χ^2. We conclude that, for the described experimental conditions, post-bagging improves classification results and that the best metric is conviction.

1 Introduction

One can use an association rule discovery strategy to obtain a large set of rules from a given dataset, and subsequently combine a subset of the rules to obtain a classification model. This two-step training process is typically heavier than building directly a model, such as a decision tree. The motivation for going the long way lies on the possibility for delaying heuristic decisions in model building, while maintaining the scalability of the process. On the other hand, association rules can be seen as Bayesian statements about the data, and can be combined using Bayesian principles in a justified way.

As an example of the power of association based classifiers we can resort to a variant of the well known XOR two class problem, with three independent attributes (x, y and z) and one dependent attribute $class$, all taking values 0 or 1. The value of $class$ is 1 if and only if x and y have different values. Attribute z introduces noise. A heuristic method, such as decision tree induction, will tend to choose z as the root variable, and then possibly fail to discover the correct answer in all the branches. A technique based on association rules can discover the 4 rules that are needed to correctly classify a new example, independently of the values of z.

* Supported by the POSI/SRI/39630/2001/Class Project (Fundação Ciência e Tecnologia), FEDER e Programa de Financiamento Plurianual de Unidades de I & D.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 137–149, 2005.

Since at least 1997, some proposals have appeared that employed association rules to obtain classification models [2] [15][17][18][20]. Such classifiers have been empirically shown as competitive in terms of predictive power (although, in the case of the works cited above, no indication of the statistical significance of the results has been provided). In this paper we explore a variant of bagging [6] to obtain a classification model from a set of association rules. In classical bagging, a number of bootstrap samples are obtained from the given training examples. For each sample, a classification model is learnt, and new cases are classified by combining the decisions of the resulting models for the new case. Bagging is therefore an ensemble method that requires a single training data set and a single model generator algorithm.

We propose and empirically evaluate a *post-bagging* method. From the training data, we obtain one set of association rules, and from that single set of rules we build a number of (partial) classification models using a bootstrap sampling approach on the set of rules.

We compare this approach with the single best rule and voting approaches using different rule characterization metrics, and also with two decision tree methods (*c4.5* and *rpart*) on 12 datasets. The empirical results provide some evidence on the average predictive power of post-bagging.

2 Classification from Association

An association rule discovery algorithm such as APRIORI [1], takes a set of transactions $D = \{T \mid T \text{ is a set of items } i\}$, a minimal support threshold σ and a minimal confidence threshold ϕ, and outputs all the rules of the form $A \rightarrow B$, where A and B are sets of items in D and $sup(A \cup B) \geq \sigma$ and $sup(A \cup B)/sup(A) \geq \phi$. $sup(X)$ is the support or the relative frequency of an item set X observed in D.

Association rule discovery can be directly applied to tabular datasets, such as the typical UCI dataset, with one column for each attibute by regarding each example as a set of items of the form $< attribute = value >$. Likewise, continuous attributes can be dealt with if discretized in advance.

Despite the fact that an association rule algorithm finds ALL rules that satisfy σ and ϕ, the discovery process can be relatively fast and discovery time grows linearly with the number of examples (clearly shown in [1] for the algorithm AprioriHybrid). This provides a scalable heuristic-free process that makes possible to avoid greedy methods such as decision trees.

The discovery of association rules can then be seen as a step preceding model building, or a computationally feasible way of having a quasi-complete search on the space of rules. A classification rule model built from such an unrestrained set of rules can potentially be more accurate than another using a greedy search approach [17,18,20].

Which is the best way of obtaining a classification model from a set of association rules is, however, not entirely clear. One can look at the set of association rules as a large decision list ordered by confidence and support [18], or by some other metric. Rules can also be combined to classify new examples through some kind of voting [17] or by using Bayesian principles [20].

In the following, we state the problem of finding a good Classification model from Association Rules.

2.1 The Problem

The problem we approach in this paper consists in obtaining a classifier, or a discriminant model M, from a set of association rules R. The rules are generated from a particular data set D of cases T, where each case T is a set of pairs $< attribute = value >$, where *value* can be categorical or numerical. One of the attributes is the *class* attribute, ranging over a finite, and typically small, set G of classes. All the rules have exactly one item on the consequent involving the *class* attribute.

We want the model M to be successful in the prediction of the classes of unseen cases taken from the same distribution as D. A Bayesian view of the success of a classifier defines that the optimal classifier M_{Bayes} maximizes the probability of predicting the correct class value $g \in G$ for a given case x [11].

$$M_{Bayes}(x) = \max_{g \in G} \Pr(g \mid x) \tag{1}$$

The success of a model M in estimating M_{Bayes} will depend on how the model is obtained from R and on how it is used to classify new cases from R. Given a case with description x, the confidence ϕ of an association rule $x' \rightarrow class = g$, with x' covering x, estimates the conditional probability $\Pr(g \mid x')$, and in the lack of more information it is a good estimator of $\Pr(g \mid x)$. The coverage relation is defined as: x covers x' iff $x' \subseteq x$ when x and x' are sets of items.

Previous work on classification from association rules has confirmed the predictive power of confidence. In this paper we provide empirical indication that another metric, *conviction*, obtains better results.

When we have a set R of association rules, we can expect to obtain more predictive power by combining different rules that apply to the same case. How to select the rules from R and how to use them is not trivial. In other words, given a rule set R, how do we obtain and use a classification model M?

3 Obtaining Classifiers from Association Rules

We can regard classification from association rules as a particular case of the general problem of model combination. Either because we see each rule as a separate model or because we consider subsets of the rules for combination. We first build a set of rules R. Then we select a subset M of rules that will be used in classification, and finally we choose a prediction strategy π that obtains a decision for a given unknown case x. To optimize predictive performance we can fine tune one or more of these three steps.

Strategy for the Generation of Rules: The simplest choice is to run APRIORI [1] once over the data D. The choice of minimal support and confidence is not trivial. Constraints on other rule charateristics can be used. A more sophisticated approach is to employ a sort of coverage strategy [18]: Build all the association rules, choose the best, remove the covered cases and repeat until all cases are covered. In [17] this standard coverage strategy is generalised to allow more redundancy between rules. A case is only removed from the training data when it is covered by a pre-defined number of rules.

In our work, we build the set of rules separately using the *Carenclass* system [4]. Carenclass is specialized in generating association rules for classification and employs a bitwise depth-first frequent patterns mining algorithm. It resembles the ECLAT algorithm proposed in [25], which is also a depth first algorithm that makes use of a vertical representation of the database.

Choice of the Rule Subset: We can use the whole set of rules for prediction, and count on the predictive strategy to dynamically select the most relevant ones. Selection of rules is based on some measure of their quality, or combination of measures. The structure of rules can also be used, for example for discarding rules that are generalizations of others. The general effort of discarding rules that are potentially irrelevant or harmful for prediction is called *pruning* [17][18].

Strategy for Prediction: Most of the previous work on using association rules for classification has been done on this topic. The simplest approach is to go for the rule with the highest quality, where quality is typically measured as the confidence of the rule, sometimes combined with support [18]. Other approaches combine the rules by some kind of *committee method*, such as simple voting [14], or weighted voting [17]. In this paper we explore another possibility inspired in *bagging* [6].

4 Rule Generation

Typically, the generation of association rules is done after the identification of frequent itemsets. For efficiency purposes, it is desirable to push the rules generation task into the frequent pattern mining phase. Frequent itemset identification is typically done as follows: first, all frequent items are identified, and then candidate itemsets are generated following an imposed order. In the case of [1] this is a lexicographic order. Other, like [25], use a support oriented order. When we interleave frequent itemset counting and rule generation, as soon as a frequent itemset is counted and checked as valid (for instance, that it contains the required consequent item), rule generation for that itemset can be triggered. However, depth-first approaches to itemset mining face a problem. It may happen that subsets of the itemset in question are not yet determined due to unfavourable ordering. Thus, we might have a rule ready to be derived (because it already contains a consequent item) but that does not have its antecedent support already counted.

Carenclass has a simple and elegant approach to this problem. Since it knows in advance which items it will generate rules for (they will occur in the consequent) it imposes an itemset ordering that keeps the itemsets involving consequent items at the end. This ensures two things: first, consequent items appear at last in an itemset; secondly, when about to generate a rule, the subset of the itemset (without the consequent item) is already counted.

5 Rule Selection

Rule selection, or pruning, can be done right after rule generation. However, most of the rule selection techniques can be used earlier when the rules are being generated.

Pruning techniques rely on the elimination of rules that do not improve more general versions. For example, rule $\{a, b, c\} \rightarrow g$, may be pruned away if rule $\{a, c\} \rightarrow g$ has similar or better predictive accuracy. CBA [18] uses pessimistic error pruning. Another possibility is to simply use some measure of *improvement* [5] on a chosen rule quality metric. Using the same example as above, if we set a minimal confidence improvement of 0.1, we may discard $\{a, b, c\} \rightarrow g$ if its confidence is less than $confidence(\{a, c\} \rightarrow g) + 0.1$. In general, $improvement(A \rightarrow B)$ can be defined as $\min(\{metric(A \rightarrow B) - metric(A_s \rightarrow B) \mid A_s \subseteq A\}$, where *metric* is a rule characterization metric such as confidence.

At modeling time we can still reduce the set of rules by choosing only the N-best ones overall, or the N-best ones for each class [14], where N is a user provided parameter. This technique may reduce the number of rules in the model dramatically, but the choice of the best value for N is not clear. The rule selection method *RC* [15] builds a decision list by traversing the generalization lattice of the rules and by looking at the training error of the rules. It starts with the most general rules, which will be at the bottom of the decision list. After that, it moves to the next level of the generalization lattice and chooses the rules that better handle the exceptions of the more general rules, while discarding the other rules at the same generalization level. This is done iteratively until the bottom of the lattice is reached.

6 Combining the Decisions of Rules

In this section we will analyze how association rules have been, and can be used for classification purposes, by studying the quality of the decisions produced. In the discussion we assume we have a static set R of classification association rules, and a predefined set of classes G and that we want to classify cases with description x, where the description of a case is a set of statements involving independent attributes. The set of rules that apply to the case, or that fire upon the case with description x will be $F(x)$ defined as $\{(x' \rightarrow class = g) \in R \mid x' \subseteq x, g \in G\}$.

Given a new case x to classify, we can use some prediction strategy to combine the rules in R.

6.1 Best Rule

This strategy tries to solve the problem with one single rule $bestrule_x$ obtained with:

$$bestrule_x = arg \max_{r \in F(x)} metric(r) \tag{2}$$

The *metric* used is a function that assigns to each rule a value of its predictive power. In this paper we study interest metrics typically used in association rule discovery: $confidence$, $conviction$, $lift$ and χ^2.

Confidence is the natural choice when it comes to prediction. It estimates the posterior probability of B given A, and is defined as $confidence(A \rightarrow B) = sup(A \cup B)/sup(A)$.

Lift is sometimes also called *interest* [8] and is a ratio between the observed support of $A \cup B$ and its expected support under the assumption that A and B are independent, $lift(A \rightarrow B) = sup(A \cup B)/(sup(A).sup(B))$. Under this assumption, the expected support is given by $sup(A).sup(B)$. Lift measures the deviation from independence of A and B. If lift is close to 1, A and B are independent, and the rule is not interesting.

Conviction is another interest metric [8] that also measures the independence of A and B, but goes a little bit further. Contrarily to lift, conviction is sensitive to rule direction ($lift(A \rightarrow B) = lift(B \rightarrow A)$). Conviction is somewhat inspired in the logical definition of implication and attempts to measure the degree of implication of a rule. Conviction is infinite for logical implications (confidence 1), and is 1 if A and B are independent.

$$conviction(A \rightarrow B) = \frac{1 - sup(B)}{1 - confidence(A \rightarrow B)} \tag{3}$$

Another way of measuring the independence of antecedent and consequent of a rule is by testing that hypothesis with a χ^2 test [19]. If the value of the statistic (equation 4) is close to zero the hypothesis of independence is accepted. How close it must be to zero depends on the level of the significance of the test.

$$\chi^2(A \rightarrow B) = \mid D \mid \sum_{X \in \{A, \neg A\}, Y \in \{B, \neg B\}} \frac{(sup(X \cup Y) - sup(X).sup(Y))^2}{sup(X).sup(Y)} \tag{4}$$

where $\mid D \mid$ is the database size.

The prediction given by the best rule is the best guess we can have with one single rule. When the best rule is not unique we can break ties maximizing support [18]. A kind of best rule strategy, combined with a coverage rule generation method, provided encouraging empirical results when compared with state of the art classifiers on some datasets from UCI [21].

However, the decision of a single rule is optimal only if we have a rule $x \rightarrow class = g$ that uses all the information in the description of the case. In general such a 'complete' rule has a very low support, most likely zero, and will not be available, or is not reliable. Therefore, we can expect to improve the quality of the prediction by using rules that use different sets of attributes in the antecedent. In [20] different rules have been combined to better approximate a Bayesian estimate of the probability of each class.

6.2 Voting

These strategies combine the rules $F(x)$ that fire upon a case x. A *simple voting* strategy takes all the rules in $F(x)$, groups the rules by antecedent, and for each antecedent x' obtains the class corresponding to the rule with highest confidence. We will denote the class voted by an antecedent x' with a binary function $vote(x', g)$ which takes the value 1 when x' votes for g, and 0 for the other classes.

$$prediction_{sv} = arg \max_{g \in G} \sum_{x' \in antecedents(F(x))} vote(x', g) \qquad (5)$$

6.3 Weighted Voting

This strategy is similar to voting, but each vote is multiplied by a factor that quantifies the quality of the vote [16]. In the case of association rules, this can be done using one of the above defined metric.

$$prediction_{wv} = arg \max_{g \in G} \sum_{x'} vote(x', g). \max metric(x' \rightarrow g) \qquad (6)$$

Carenclass implements these and other prediction strategies efficiently by keeping in an appropriate data structure [3].

In the next section we describe a technique for rule combination inspired in bagging.

7 Bagging Association Rules for Classification

Bagging is the generation of several models from bootstrap samples of the same original dataset D [6]. The prediction given by the set of resulting models for one example e is done by averaging the predictions of the different models. Bagging has the effect of improving the results of an unstable classifier by reducing its variance [11]. Domingos [9] suggests that, in the case of decision trees, bagging works because it increases the probability of choosing more complex models.

In the case of classification from association, we obtain a large set of rules R that contain many alternative possible models. So what we propose is the technique we call *post-bagging.* It consists in sampling repeatedly the set of rules a posteriori to obtain an ensemble of models similarly to bagging. The models in a particular ensemble will be similar, but their differences will tend to reflect the variability of rule sets obtained from the same source of data.

New cases are classified by obtaining the prediction of each of the models in the ensemble (and this can be done with any strategy), and using simple voting to combine those predictions. Experimental evaluation indicates that this technique can obtain good results when compared to a bestrule or a voting approach, or even to decision tree learners, such as *c4.5* [23] and *rpart* [13].

We will now describe the BAGGAR (Bootstrap Aggregation of Association Rules) algorithm (Algorithm 1) in detail. After obtaining a set of association rules R from a dataset D, we build a number of *baggs* from R. Each bagg is a sample with a pre-defined size of the rule set. Sampling is performed with replacement. The number of baggs (*n.baggs*) is 30 by default, and the size T of each bag is, in general, 10%. These defaults have been set in preliminary experiments and should not be regarded as necessarily ideal.

Algorithm 1. *Baggar Algorithm, training*
Given: a set E with labelled examples; *n.bags*: the number of baggs (default 30); T: size of each bag (default $\min(|R|, \max(50, 0.1\times |R|))$)
Do:

```
1. Build a set $R$ of Association Rules
2. For $i$ in 1 to $n.bags$
3.     $S_i \leftarrow$ sample with replacement from $R$ of size $T$
```

Output: the set of baggs $\{S_i\}$

The classification of a single example e using a set of baggs $\{S_i\}$ is done by applying a chosen prediction strategy π to each of the baggs. The most voted class is then output as the overall prediction.

8 Empirical Evaluation

To test the value of post-bagging, we have compared different variants of *carenclass*, corresponding to different prediction strategies, on 12 UCI datasets [21]. To serve as a state of the art reference, we used the decision tree inducer *c4.5* [23]. Due to its availability and ease of use we have also compared the results with *rpart* from the statistical package *R* [13]. *Rpart* is a CART-like decision tree inducer [7].

We used eight carenclass variants, by combining two strategies: "Best rule" and "Weighted Voting" with four metrics (confidence, conviction, lift and χ^2). Minimal support was set to 0.02 and minimal improvement to 0.01. For each combination we ran carenclass with and without post-bagging. Numerical attributes have been previously discretized using Weka's [24] implementation of Fayyad and Irani's supervised discretization method [10].

An estimation of the error of each algorithm (and carenclass variant) was obtained on each dataset with a 10×10-fold cross-validation (Table 2). From the estimated errors we ranked the algorithms separately for each dataset, and used mean ranks as an indication of global rank. Besides that, we have studied the statistical significance of the results obtained.

8.1 Post-Bagging Ranks High

The first empirical observation is that 3 post-bagging variants rank high among the four top places (Table 3). Compared to c4.5 and rpart, 5 carenclass variants

Table 1. Datasets used for the empirical evaluation

Dataset	#examples	#classes	#attr
australian	690	2	14
breast-wisconsin	699	2	9
cleveland	303	5	13
diabetes	768	2	8
flare	1066	2	10
heart	270	2	13
hepatitis	155	2	19
house votes	435	2	16
german	1000	2	20
segment	2310	7	19
vehicle	846	4	18
yeast	1484	10	8

Table 2. Average error rates obtained with the algorithms on the datasets (minimal support=0.02 and improvement=0.01)

	austr	breas	diabe	flare	cleve	yeast	house	germa	vehic	heart	hepat	segme
rpart	0.1504	0.0611	0.2572	0.1831	0.4566	**0.4276**	0.0481	0.2611	0.3213	0.1856	0.3044	0.0831
c4.5	0.1493	0.0512	0.2599	**0.1804**	0.4939	0.4408	**0.0343**	0.2862	**0.2690**	0.2205	0.2122	**0.0324**
Bestrule.conf	0.1432	0.0438	0.2279	0.1884	0.4587	0.4439	0.0770	0.2961	0.3968	0.1767	0.1794	0.1731
Bestrule.lift	0.3096	0.1890	0.4158	0.2179	0.6545	0.5237	0.4457	0.5865	0.4358	0.2270	0.5819	0.1752
Bestrule.conv	0.1409	0.0413	0.2236	0.2039	0.4466	0.4408	0.0770	0.2801	0.3961	0.1715	0.1794	0.1729
Bestrule.chi	0.1449	0.0635	0.2798	0.1978	0.4409	0.4558	0.0498	0.3059	0.4893	0.2522	0.3006	0.3315
Voting.conf	0.1800	0.0372	0.2301	0.1857	0.4306	0.4591	0.1236	0.2558	0.3590	0.1759	0.2314	0.2808
Voting.lift	0.1686	**0.0300**	0.2365	0.1936	0.4565	0.4448	0.1417	0.2592	0.3586	0.1707	0.3663	0.2805
Voting.conv	0.1542	0.0376	0.2244	0.1856	0.4272	0.4453	0.1101	**0.2465**	0.3437	**0.1596**	0.2108	0.1971
Voting.chi	0.1448	0.0388	0.2400	0.1913	0.4285	0.4397	0.1423	0.2683	0.3872	0.1767	0.2401	0.2914
Bag.Bestrule.conf	0.1351	0.0378	0.2271	0.1880	0.4345	0.4480	0.0723	0.2883	0.3696	0.1696	**0.1663**	0.2533
Bag.Bestrule.lift	0.2035	0.0571	0.3278	0.2104	0.5065	0.4764	0.2767	0.4322	0.3831	0.1681	0.5099	0.2515
Bag.Bestrule.conv	**0.1345**	0.0329	0.2246	0.1984	0.4361	0.4426	0.0739	0.2699	0.3702	0.1648	0.1672	0.2533
Bag.Bestrule.chi	0.1480	0.0499	0.2582	0.1968	**0.4176**	0.4488	0.1457	0.2961	0.4186	0.1800	0.2554	0.3196
Bag.Voting.conf	0.1810	0.0376	0.2283	0.1853	0.4326	0.4582	0.1220	0.2562	0.3597	0.1737	0.2314	0.2819
Bag.Voting.lift	0.1703	**0.0300**	0.2381	0.1939	0.4518	0.4427	0.1393	0.2567	0.3589	0.1707	0.3622	0.2823
Bag.Voting.conv	0.1394	0.0342	**0.2219**	0.1850	0.4287	0.4469	0.0778	0.2528	0.3437	0.1659	0.2048	0.2592
Bag.Voting.chi	0.1535	0.0399	0.2424	0.1906	0.4318	0.4435	0.1425	0.2636	0.3876	0.1778	0.2414	0.3043

rank higher than those. Although this is a good indication of the predictive power of post-bagging, we still have to discriminate its effect from the effect of the metric, and test its statistical significance.

To perceive the specific effect of post-bagging, we can observe that 5 (Voting.conv, Bestrule.conv, Bestrule.conf, Bestrule.chi and Bestrule.lift) against 3 (Voting.conf, Voting.lift, Voting.chi) of the carenclass variants benefit from post-bagging. The improvement is more visible on the simple Bestrule approach, rather than Voting. This may be explained by the fact that Voting is already a multi rule method.

We should note that the segment data set appears as a particularly difficult task for our association rule approaches. This is the data set, out of these 12, where the tree approaches perform visibly better. Moreover, it is also the only data set where post-bagging does not improve the results of best rule with confidence as a metric. In fact, post-bagging obtains very bad results. The segment data set has seven equally balanced classes. However, the number of rules per class tend to be unbalanced, which may be the reason for the higher error of the

Table 3. Ranks obtained (minimal support=0.02 and improvement=0.01)

	mean	austr	breas	diabe	flare	cleve	yeast	house	germa	vehic	heart	hepat	segme
Bag.Voting.conv	4.79	3	4	1	3	4	11	8	2	3.5	3	5	10
Voting.conv	5.42	12	6.5	3	5	2	10	9	1	3.5	1	6	6
Bag.Bestrule.conv	6.21	1	3	4	15	9	5	5	10	10	2	2	8.5
Bag.Bestrule.conf	6.88	2	8	5	7	8	12	4	13	9	5	1	8.5
Bestrule.conv	7.79	4	11	2	16	11	3.5	6.5	11	14	8	3.5	3
c4.5	8.04	9	14	15	1	16	3.5	1	12	1	16	7	1
rpart	8.17	10	16	13	2	14	1	2	7	2	15	14	2
Voting.conf	8.88	15	5	8	6	5	16	11	3	7	10	8.5	12
Bag.Voting.conf	9	16	6.5	7	4	7	15	10	4	8	9	8.5	13
Bestrule.conf	9.08	5	12	6	8	15	8	6.5	14.5	15	11.5	3.5	4
Voting.chi	9.38	6	9	11	10	3	2	14	9	12	11.5	10	15
Voting.lift	9.5	13	1.5	9	11	13	9	13	6	5	6.5	16	11
Bag.Voting.lift	9.5	14	1.5	10	12	12	6	12	5	6	6.5	15	14
Bag.Voting.chi	10.92	11	10	12	9	6	7	15	8	13	13	11	16
Bag.Bestrule.chi	12.62	8	13	14	13	1	13	16	14.5	16	14	12	17
Bestrule.chi	13.67	7	17	16	14	10	14	3	16	18	18	13	18
Bag.Bestrule.lift	14.42	17	15	17	17	17	17	17	17	11	4	17	7
Bestrule.lift	16.75	18	18	18	18	18	18	18	18	17	17	18	5

best rule approach: classes with more rules tend to get more votes. Similarly, the rule distribution per class is unbalanced in the produced baggs.

8.2 Conviction Ranks High

Four of the five top places are taken by carenclass variants that use conviction as a rule value metric. Moreover, conviction always has higher mean ranks than all the other metrics with respect to all the variants, and always higher than c4.5 and rpart. This predictive power of conviction is somewhat surprising and deserves to be better explained in the future. One possibility for the apparently good predictive performance of this metric may be due to the fact that it tends to favour less frequent classes. In particular, given two rules with the same confidence, conviction prefers the one whose consequent has lower support (Equation 3). The results with segment corroborate this intuition. This is a problem with 7 equally frequent classes. As a result, confidence and conviction practically have the same results. This is also observed on datasets with an almost balanced number of classes (australian, heart, vehicle).

The second best metric is clearly confidence. χ^2 and lift seem more or less equivalent in terms of results with a slight advantage to χ^2. Note that these two metrics are symmetric w.r.t. antecedent and consequent of the rule, contrarily to confidence and conviction.

8.3 Statistical Significance of Results

Although the average ranks provide a good overall picture of the results, these should be verified in terms of statistical significance. Our claims are based on statements of the form "algorithm x is better than algorithm y", and "the tested algorithms perform equally". To assess the statistical significance to such statements we will use paired *t-tests* and the *Friedman rank sum* test [22]. The *t-tests* are used as follows. For each partition of a dataset, we average the 10 error values obtained with a given algorithm from 10-fold cross validation. Since we have 10 different partitions we obtain 10 average errors. To compare two algorithms we perform an hypothesis test where the two samples are the average errors of each algorithm on the same dataset. The null hypothesis is that the algorithms perform equally. The alternative hypothesis is accepted if the p value for the t statistic is lower than 0.001. *Friedman* tests the hypothesis that all the methods have equal performance.

Table 4. Statistically significant ($\alpha = 0.001$) wins

	Bestrule				Voting			
	conv		conf		conv		conf	
	Bagging	Simple	Bagging	Simple	Bagging	Simple	Bagging	Simple
c4.5	3/4	4/3	3/5	3/2	3/5	3/5	4/5	5/5
rpart	3/3	4/3	5/3	4/3	3/3	2/3	5/3	5/3

If we compare directly post bagging and single model variants using *t-test*, we observe that statistically significant wins are not outstanding (Table 4). We mostly observe a near-draw with a slight advantage in favour of the post-bagging variants. However, if we separately compare post-bagging and the respective single model variant with c4.5, we observe a higher number of statistically significant wins of the post-bagging approach. With respect to rpart, post-bagging tends to improve the results of the single model variants. Compared with post-bagging, we observe an advantage of rpart, despite the fact that the direct comparison between rpart and c4.5 is favourable to the latter (4 wins against 2).

By using Friedman's test on all the data on Table 2, we reject the hypothesis that all the approaches have equal performance with very high confidence (p-value is lower than 10^{-7}). However, if we take out the carenclass variants that use lift and χ^2, p-value goes up to 0.13. Despite the good indications given by the ranking and the t-tests, and despite the fact that similar rankings are observed when the parameters of post-bagging are changed (number of baggs=30, 50, 70, 200; size of baggs=50%, minsup=0.01), we cannot firmly claim that there is a highly significative advantage in using post-bagging.

9 Conclusions

We have presented the technique of post-bagging, which produces an ensemble of rule classification from a single set of association rules. Post bagging has the advantage that a single model is built from the dataset and bootstrap models are built from this one. Empirical experiments indicate that post-bagging outranks on average standard decision tree techniques and tends to improve the results of bestrule, for the metrics considered. The effect of post-bagging on voting is only marginally positive, using confidence and conviction. We hypothesize that this is probably due to the fact that voting is already a multi rule decision method, and post-bagging has little room for improvement.

In terms of metrics, conviction tends to give better results than confidence, which is the second best metric. This is probably because class frequency is taken into account by conviction but not by confidence. The other two metrics (lift and χ^2) have been included for the sake of completeness but are far from being competitive.

We also observe that a simple Bestrule approach (generate rules-use best rule) gives competitive results: slightly better than c4.5 with conviction, slightly worse with confidence.

In conclusion, we can say that it is worthwhile to proceed with the research on post-bagging, and to better study the reasons for failure and success according to data set and rule set characteristics (number of classes, class distribution, number of rules, number of rules per class). This could lead us to improved classification accuracy and a better insight of the classification problem itself.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A. I.: Fast Discovery of Association Rules, Advances in Knowledge Discovery and Data Mining: 307-328, (1996).
2. Ali, K., Manganaris, S. and Srikant, R.: Partial classification using association rules, Proceedings of the Third ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-97, ACM, 115-118, (1997).
3. Azevedo, P.J., A Data Structure to Represent Association Rules based Classifiers Technical Report, Universidade do Minho, (2005).
4. Azevedo P.J., Jorge, A.M., The CLASS Project, http://www.niaad.liacc.up.pt/~amjorge/Projectos/Class/
5. Bayardo, R.J., Agrawal, R., Gunopulos, D., Constraint-Based Rule Mining in Large, Dense Databases, Data Mining and Knowledge Discovery, Volume 4, Issue 2 - 3, Pages 217 - 240, (2000)
6. Breiman, L.: Bagging Predictors, Machine Learning, Vol. 24, No. 2, pp. 123-140, (1996).
7. Breiman, L., Friedman, J.H., Olshen, R. A., Stone, C. J. : Classification and Regression Trees. Wadsworth, (1984).
8. Brin, S., Motwani, R., Ullman, J. D., and Tsur,S., Dynamic itemset counting and implication rules for market basket data, Proceedings of th ACM SIGMOD International Conference on Management of Data, (1997).
9. Domingos, P., Why does bagging work? A Bayesian account and its implications, Proceedings of the Third ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-97, ACM, 115-118, (1997).
10. Fayyad, U.M., Irani, K. B.: Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, in Proceedings of the 13th International Joint Conference on Artificial Intelligence, R. Bajcsy (Ed.): Chambry, France, Morgan Kaufmann, pp. 1022-1029, (1993).
11. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference and Prediction, Series in Statistics, Springer, (2001).

12. Ho, T. K., Hull, J. J., Srihari, S. N.: Decision Combination in Multiple Classifier Systems, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(1), 66-75, (1994).
13. Ihaka, R., and Gentleman, R.: R: A Language for Data Analysis and Graphics, Journal of Computational Graphics and Statistics,Vol. 5, N. 3, pp. 299-314, (1996).
14. Jovanoski, V., Lavrac, N.: Classification rule learning with APRIORI-C, in Proc. of EPIA 2001, P. Brazdil, A. Jorge (Eds.), Springer, LNCS 2258, 44-51, (2001).
15. Jorge, A., Lopes, A.:Iterative Part-of-Speech Tagging, Learning Language in Logic, J. Cussens, S. Dzeroski (Eds), LNAI 1925, Springer-Verlag, (2000).
16. Kononenko, I.: Combining decisions of multiple rules, Artificial Intelligence V: Methodology, Systems, Applications, B. du Boulay, V. Sgurev (Eds.), Elsevier (1992).
17. Li, W., Han, J., Pei, J.: CMAR: Accurate and Efficient Classification Based on MultipleClass-Association Rules, in IEEE International Conference on Data Mining, (2001).
18. Liu, B., Hsu, W. e Ma, Y.: Integrating Classification and Association Rule Mining, Proceedings of the Fourth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 15-18, 1998, New York, USA. ACM, (1998).
19. Liu, B., Hsu, W. , Ma, Y., Pruning and Summarizing the Discovered Associations, Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 15-18, 1999, San Diego, CA, USA. ACM, 125-134, (1999).
20. Meretakis, D., Wüthrich, B.,: Extending Nave Bayes Classifiers Using Long Itemsets, Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 15-18, 1999, San Diego, CA, USA. ACM, pp. 165-174, (1999).
21. Merz, C. J., Murphy, P.:UCI Repository of Machine Learning Database. http://www.cs.uci.edu/∼mlearn, (1996).
22. Neave, H.R., Worthington,P.L.: Distribution-free tests, Unwin Hyman Ltd. , (1988).
23. Quinlan, J. R.: C4.5: Programs for Machine Learning, Morgan Kaufmann, (1993).
24. Witten, I. H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, (1999).
25. Zaki, M.J., Scalable algorithms for association mining, IEEE Transactions on Knowledge and Data Engineering, 12(3):372-390, (2000).

Movement Analysis of Medaka (*Oryzias Latipes*) for an Insecticide Using Decision Tree

Sengtai Lee[1], Jeehoon Kim[2], Jae-Yeon Baek[2], Man-Wi Han[2], Chang Woo Ji[3], and Tae-Soo Chon[3]

[1] School of Electrical Engineering, Pusan National University,
Jangjeon-dong, Geumjeong-gu, 609-735 Busan, Korea
youandi@pusan.ac.kr
[2] Korea Minjok Leadership Academy,
Sosa-ri, Anheung-myeon, Heongseong-gun, Gangwon-do, 225-823, Korea
{fantasy002, mrswoolf}@hanmail.net, manwihan@chol.com
[3] Division of Biological Sciences, Pusan National University,
Jangjeon-dong, Geumjeong-gu, 609-735 Busan, Korea
tschon@pusan.ac.kr

Abstract. Behavioral sequences of the medaka (*Oryzias latipes*) were continuously investigated through an automatic image recognition system in response to medaka treated with the insecticide and medaka not treated with the insecticide, diazinon (0.1 mg/l) during a 1 hour period. The observation of behavior through the movement tracking program showed many patterns of the medaka. After much observation, behavioral patterns were divided into four basic patterns: active-smooth, active-shaking, inactive-smooth, and inactive-shaking. The "smooth" and "shaking" patterns were shown as normal movement behavior. However, the "shaking" pattern was more frequently observed than the "smooth" pattern in medaka specimens that were treated with insecticide. Each pattern was classified using a devised decision tree after the feature choice. It provides a natural way to incorporate prior knowledge from human experts in fish behavior and contains the information in a logical expression tree. The main focus of this study was to determine whether the decision tree could be useful for interpreting and classifying behavior patterns of the medaka.

1 Introduction

Ecological data are very complex, unbalanced, and contain missing values. Relationships among variables may be strongly nonlinear and involve high-order interactions. The commonly used exploratory and statistical modeling techniques often fail to find meaningful ecological patterns from data [1], [2], [3]. The behavioral or ecological monitoring of water quality is important regarding bio-monitoring and risk assessment [4], [5]. An adaptive computational method was utilized to analyze behavioral data in this study. The decision tree is a modern statistical technique that is ideally suited for both exploring and modeling data. It is constructed by repeatedly splitting the data, and defined by a simple rule based on a single explanatory variable.

Recently, behavioral responses to sub-lethal doses of toxic chemicals have drawn attention as a means of developing a bio-monitoring tool for detecting toxic chemicals in the environment. Behavioral responses have been reported to be sensitive to sub-

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 150 – 162, 2005.

lethal exposures to various chemical pollutants [4], [5]. In recent years, research on the effects of sub-lethal levels of toxic substances has been rapidly accumulating for various taxa, including crustaceans [6], snails [7], fish [8], [9], and insects [10]. However, these studies are mostly based on observation of single or combinations of single behaviors with qualitative descriptions. Not much quantitative research has been conducted on behavioral changes in spatial and temporal domains in response to treatments of toxic chemicals.

The observation of the movement tracks of small sized animals has been separately initiated in the field of search behavior in chemical ecology [11] and computational behavior [12], [13]. In regard to searching behavior, the servometer and other tools were used for investigating the continuous movement tracks of insects, including cockroaches, in characterizing the effects of wind [14], pheromone [15], [16], relative humidity [17], and sucrose feeding [18].

In regard to the computational aspects [12], Alt modeled the movement of the organism, such as the circling path of gametes or the meander search by isopods, and Scharstein [19] revealed a complex directional autocorrelation function with monotonic decay and discontinuity at the origin. Tourtellot *et al.* [13] analyzed the movement length and turn definition in the analysis of the orientation data of cockroaches. Johnson *et al.* and Weins *et al.* attempted to quantify insect movements and suggested the fractal dimensions of pathway configurations [20], [21]. Recently, studies on rats were conducted in dynamic perspectives and statistical discrimination of motion in exploration behavior [22], [23].

These computational methods convey useful mathematical information regarding similarities present in the data of the movement tracks; for instance, correlation coefficients or fractal dimensions. Using these methods, however, the parameters are obtained through mathematical transformations of the movement data, and information is in a generally highly condensed state. These methods are usually not interpretable for uniquely and directly characterizing the actual shape of the movement tracks.

In this paper, we utilized the decision tree for the classification of response behaviors and attempted to explain the shapes of the movement tracks through feature extraction in response to sub-lethal treatments of an insecticide. The decision tree is a widely used technique for data classification and prediction. One of its advantages is that rules, which are easy to understand, can be induced. Realizing that there is a limit to observing with the naked eye, computational methods were used to conduct our research more effectively. First, statistical analysis in total moving distance, average speed, and sectional domination was conducted as a feature extraction. Furthermore, we devised a new analysis method for pattern isolation based on a decision tree to differentiate the patterns we thought distinctive. This research can help the biosensor field in detecting defects in fish, or in finding out chemical toxicants that exist in the water by observing specific behavior patterns of fish.

2 Experiments for Data Acquisition

The specimens of medaka (*Oryzias latipes*) used in our experiment were obtained from the Toxicology Research Center, Korea Research Institute of Chemical Technology (KRICT; Taejon, Korea). Only the specimens six to twelve months old were used. The medaka is about 4cm in length and lives for about 1-2 years. Because it is an easy species to rear and reproduce, it is used widely in the research of genetics or as a testing material in the detection of water pollution.

Before the experiment, specimens of medaka were maintained in a glass tank and were reared with an artificial dry diet (Tetramin^{TM}, Tetra Werke, Germany) under the light regime of Light 10: Dark 14 at a temperature of 25 ± 0.5˚C. The water in the tank was continually oxygenated prior to experimentation.

A day before experimentation, the medaka was put into the observation tank and was given approximately twelve hours to adjust. The specimens were kept in a temperature of 25˚C and were given a sufficient amount of oxygen during these twelve hours prior to the experiment. The specimens used were male and about 4cm long. In order to achieve image processing and pattern recognition effectively, a stable condition was maintained in the monitoring system. Disturbances to observation tanks and changes in experimental conditions were minimized. Aeration, water exchange and food were not provided to the specimens during the observation period and the light regime was kept consistent.

The aquarium used was based on the experiments on fish behavior in KRICT. The observed aquarium size was 40cm×20cm×10cm. Diazinon (DongYang Chemical; O,O-diethyl O-2-isopropyl-4-methyl-6-pyrimidyl thiophosphate, 93.9%) dissolved in dimethylsulfoxide (DMSO; 10mg/l), was introduced at the concentration of 0.1mg/L directly into an aquarium in which a 6-12 month old individual adult medaka specimen resided. During the period of observation, individual medaka specimens were placed in a glass aquarium. The analog data captured by the camera set in front of the aquarium were digitized by using the video overlay board every 0.25 seconds and were sent to the image recognition system to locate the target in spatial time domains. The spatial position of the medaka was recorded in two-dimensional *x*, *y* coordinate values. After giving the experimenting specimen approximately twelve hours to adjust to the observation aquarium, the experiment was started. The experiment was started at about 8:00~8:30 AM every day. Each data from a movement pattern had an interval of one minute.

3 Feature Extraction Process

3.1 Images of Movement Behavior of Medaka

In this paper, the movement patterns of the medaka were classified into shaking and smooth patterns as shown respectively in Fig. 1 and 2. When a medaka was affected

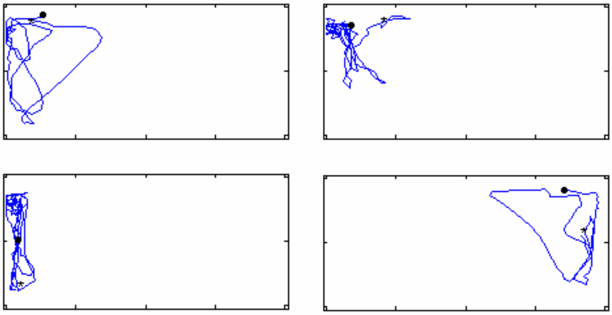


Fig. 1. Example of shaking patterns during a one-minute interval (•: start, *: end)

y diazinon (0.1 mg/l), the treated specimen was generally less active, and the movement behavior was shaky and interspersed with irregular, repetitive back-and-forth movements. Response behaviors were frequently vertical as can be observed in Fig. 1,

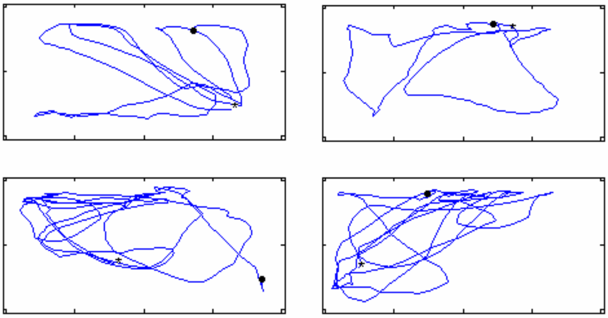


Fig. 2. Example of smooth patterns in a one-minute interval (•: start, *: end)

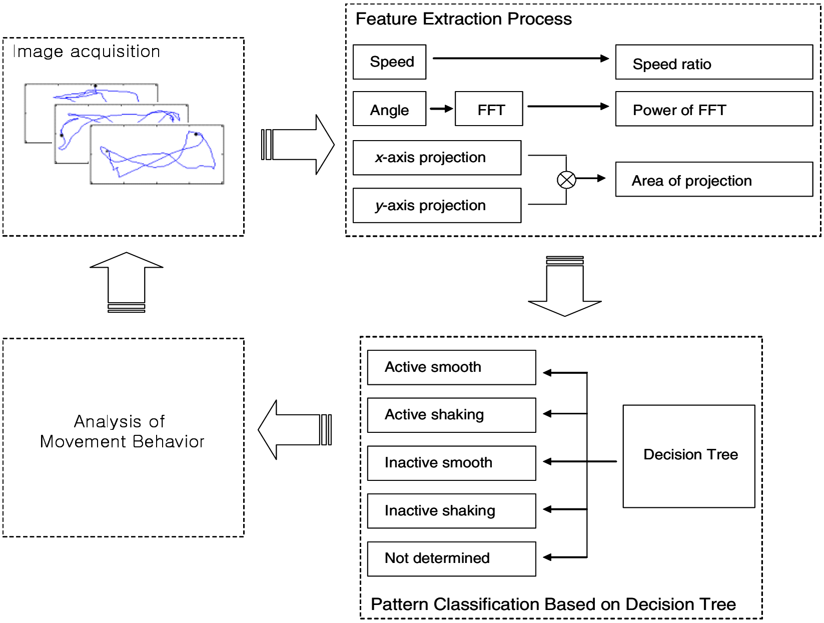


Fig. 3. Schematic diagram for automatic pattern isolation

and the degree of shaking was usually higher during the upward climb. Fig. 2 shows generally not be treated species. The behavior of the medaka in a minute period of time was used to classify them into 5 patterns: active-smooth, active-shaking, inactive-smooth, inactive-shaking, and not determined in each case. “Not determined” means that a pattern was not classified into any one of these four categories. By the observation of an expert in fish behavior to initiate pattern isolation, the features were observed and the following three feature variables could be defined: high-speed ratio, FFT (Fast Fourier transformation) to angle transition, and projection to *x*- and *y*-axes. Fig. 3 shows the schematic diagram during one minute of the movement analysis for the process of extracting three distinctive characteristics from the data acquired and classifying 5 patterns based on this information. It is possible that some patterns may not have been classified for medaka treated with sub-lethal chemicals. However, in these cases, further analysis and observation can add new patterns and update the decision tree.

3.2 Feature Extraction from Images

In order to know the activeness of a medaka, speed information was used to define high-speed ratio. The speed of the medaka shows whether the pattern is an active movement or inactive movement. The formula for speed is as the following:

$$S = \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2} \quad n = 1, 2, 3, \cdots \tag{1}$$

Here, x_n and y_n are the position values of the medaka in a sampled time. The ratio of patterns that exceeded the calculated average speed of the overall 7 data sets, 21mm/sec and the total number of patterns was used as the first feature variable. High-speed ratio is calculated using the following equation. A2 represents the average speed of the overall 7 data sets in equation (2).

$$S_{ratio} = \frac{\text{Number of samples above A2}}{\text{Number of samples in one minute}} \times 100(\%) \tag{2}$$

The change of direction in the movement track was observed to consider movements of medaka. The change of direction is represented as an angle transition to classify the movement. Angle transition between two sampled times denoted as *H* is calculated in the following equation. Here x_n and y_n show the coordinate value for the *x* and *y* axes.

$$H = \arctan(\frac{y_{n+1} - y_n}{x_{n+1} - x_n}), \qquad n = 1, 2, \cdots \tag{3}$$

Fourier transformation is used to transform signals in the time domain to signals in the frequency domain [30]. We apply the Fast Fourier Transform (FFT) to the signal of angle transition in order to calculate energy. The FFT for a given discrete signal *x*[*n*] is calculated using the following equation:

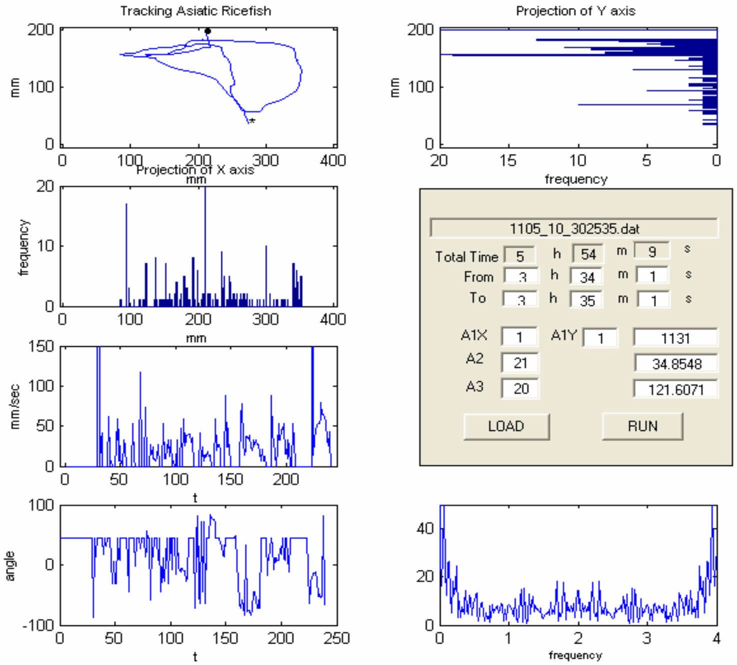
$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j(2\pi kn/N)}, \qquad k = 0, 1, \cdots, N-1. \tag{4}$$

After applying the FFT to angle transition, the power of FFT (PF) is calculated using the following equation for the amplitudes above a median.

$$P = \sqrt{\sum_{i=1}^{k} x_i^2} \tag{5}$$

Here, x_i is the amplitude above a median. We use all sets to find the median in experiments. We are used to FFT power because of the calculation in qualified angle transition. The PF is employed as the second feature variable for pattern isolation.

Projection is a method of showing a shape in a two-dimensional graph into a shape in a one-dimensional graph. In this paper, the method of projection was used to observe and understand the movement route of the medaka in a two-dimensional space. The projection to the *x*-axis and the projection of the *y*-axis were calculated and then multiplied to figure out the area of the movement track of the medaka. The calculated area tells whether the medaka moved broadly all over the tank or in a restricted area of the tank. The area calculated was used as the third variable to classify patterns.



1105_10_302535.dat

Total Time 5 h 54 m 9 s

From 3 h 34 m 1 s

To 3 h 35 m 1 s

A1X 1 A1Y 1 1131

A2 21 34.8548

A3 20 121.6071

Fig. 4. Movement tracking program

Fig. 4 shows a program that was devised using Matlab environment to analyze the data that was acquitted consecutively. This picture shows the path of the medaka in a two-dimensional space and the three numerical values for the distinctive characteristics mentioned above.

4 Classification Based on Decision Tree

4.1 Decision Tree

A decision tree is a graph of decisions and possible consequences, used to create a plan to reach a goal. Decision trees are constructed in order to help make decisions. It has interpretability in its own tree structure. Such interpretability has manifestations which can easily interpret the decision for any particular test pattern using the conjunction of decisions along the path to its corresponding leaf node. Another manifestation can occasionally get clear interpretations of the categories themselves, by creating logical descriptions using conjunctions and disjunctions [3], [29].

Many people related to artificial intelligence research has developed a number of algorithms that can automatically construct decision tree out of a given number of cases, e.g. CART [1], ID3 [24], [25], C4.5 [26], [27], [28]. The C4.5 algorithm is the most popular in a series of "classification" tree methods. In the C4.5 algorithm, real-valued variables are treated in the same as in CART.

A decision tree consists of nodes(*N*) and queries(*T*). The fundamental principle underlying tree creation is simplicity. We prefer decisions that lead to a simple, compact tree with few nodes. During the process of building the decision tree, we seek a property query *T* at each node *N* that makes the data reaching the immediate descendent nodes as "pure" as possible. It turns out to be more convenient to define the impurity, than to define the purity of a node. Several different mathematical measures of impurity have been proposed, i.e. entropy impurity (or occasionally information impurity), variance impurity, *Gini* impurity, misclassification impurity are shown in equations (6), (7), (8), (9) respectively.

$$i(N) = -\sum_j P(\omega_j)\log_2 P(\omega_j) \tag{6}$$

Here $i(N)$ denotes the impurity of a node and $P(w_i)$ is the fraction of patterns at node *N* that are in category w_j.

$$i(N) = P(\omega_1)P(\omega_2) \tag{7}$$

$$i(N) = \sum_{i \neq j} P(\omega_i)P(\omega_j) = 1 - \sum_j P^2(\omega_j) \tag{8}$$

$$i(N) = 1 - \max_j P(\omega_j) \tag{9}$$

All of them show basically the same behavior. Based on the well-known properties of entropy, if all the patterns are of the same category, the entropy impurity is 0. A variance impurity is particularly useful in the two-category case. A generalization of the variance impurity, applicable to two or more categories, is the *Gini* impurity in equation (8). This is just the expected error rate at node *N* if the category label is selected randomly from the class distribution present at *N*. The misclassification impurity measures the minimum probability that a training pattern would be misclassified at *N*. Out of all the impurity measures typically considered, this measure is the most strongly peaked at equal probabilities.

In order to drop in impurity, we used the equation (10)

$$\Delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R) \tag{10}$$

Here N_L and N_R are respectively the left and right descendent nodes, $i(N_L)$ and $i(N_R)$ are their impurities, and P_L is the fraction of patterns at node *N* that will go to N_L when property query *T* is used. Then the "best" query value *s* is the choice for *T* that maximizes $\Delta i(T)$.

If we continue to grow the full tree until each leaf node corresponds to the lowest impurity, the data have been typically overfitted. Conversely, if splitting is stopped too early, then the error on the training data is not sufficiently low and performance may suffer. To search for the sufficient splitting value, we used cross-validation (10-fold cross validation). In cross-validation, the tree is trained using a subset of the data with the remainders kept as a validation set.

4.2 Implementation of Decision Tree

We analyzed movement tracks of the medaka using Matlab6.1. The decision tree is employed and programmed to express the classification in the form of a tree and as a set of *IF-THEN* rules.

In order to classify the patterns into active smooth, active shaking, inactive smooth and inactive shaking divided by the experts in fish behavior, the following features were used: high speed ratio (HSR), power of FFT (PF), and area of projection product (APP). These 3 features were used as input variables for the decision tree. The training data for the decision tree consisted of 30 data in each pattern: active smooth, active shaking, inactive smooth, inactive shaking, and not determined.

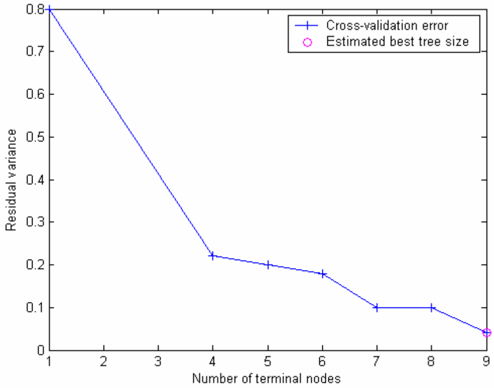


Fig. 5. Result of cross-validation (hold-out method)

We continue splitting nodes in successive layers until the error on the validation data is minimized. Cross validation is a model evaluation method that is better than residuals. The problem with residual evaluations is that they do not give an indication of how well the learner will do when it is asked to make new predictions for data it has not already seen. One way to overcome this problem is to not use the entire data set when training a learner. Some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model. We used a 10-fold cross validation for model evaluation. The data set of the 10-fold cross validation is divided into 10 subsets, and the holdout method is repeated 10 times. Result of this method is shown in Fig. 5.

The principal alternative approach to stopped splitting is pruning. Fig. 6 shows the decision tree applied to evaluated pruning. This benefit of this logic is that the reduced rule set may give improved interpretation.

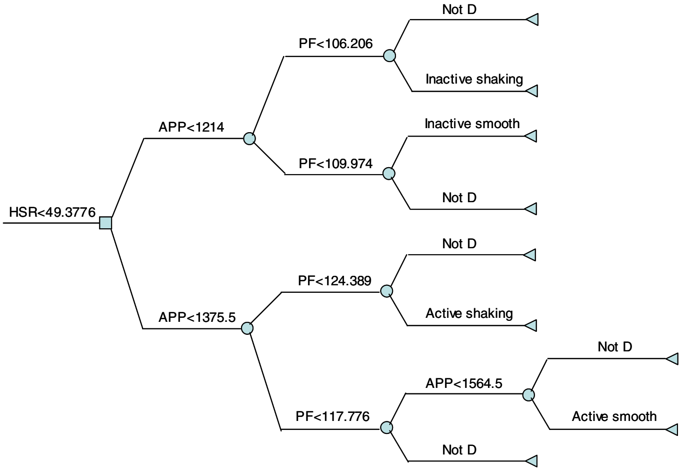


Fig. 6. The decision logic for pattern classification generated by decision tree applied to pruning. (HSR: high-speed ratio, APP: area of projection product, PF: power of FFT).

5 Behavior Analysis and Discussion

5.1 Analysis of Movement Behavior

The decision tree was applied into the movement tracks of the medaka at real-time to classify patterns. We developed models based on the classification and regression tree (CART) in order to classify and recognize movement tracks of the medaka for an insecticide. Matlab6.1 was used in order to create the program. Results were calculated for the decision logic for 60 minutes. The specimens used in the experiment were 10 medakas treated with insecticide and 10 medakas not treated with insecticide. The recognition is calculated by 5 patterns that includes “not determined.” “Smooth” means that “active smooth” patterns and “inactive smooth” patterns appeared in the decision tree logic. “Shaking” means that “active shaking” patterns and “inactive shaking” patterns appeared in the decision tree logic. “Not determined” means that neither “smooth” nor “shaking” appeared in the decision tree logic.

Fig. 7 shows the ratio of “smooth”, “shaking” and “not determined” patterns in specimens without the insecticide. “Smooth” is the sum of active smooth and inactive smooth patterns. “Shaking” is the sum of active shaking and inactive shaking. “Not” is the sum of patterns that are not inactive shaking, not inactive smooth, not active shaking, and not active smooth shown in Fig. 6. Among the 10 data sets, the recognition rate of specified patterns in smooth and shaking is in a range of 48~77% while the rate of average is 62.1%. Most specimens showed more smooth patterns detected by the decision tree logic.

Fig. 8 shows the ratio of “smooth”, “shaking” and “not determined” patterns in medaka that were treated with medaka. Among the 10 data sets, the recognition rate of specified patterns in smooth and shaking is 23~57% and the recognition rate of

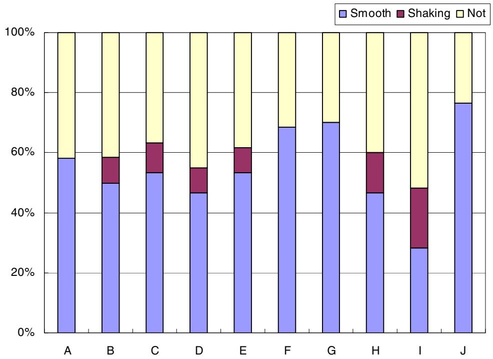


Fig. 7. Recognition rate for each pattern in set of medaka without the insecticide

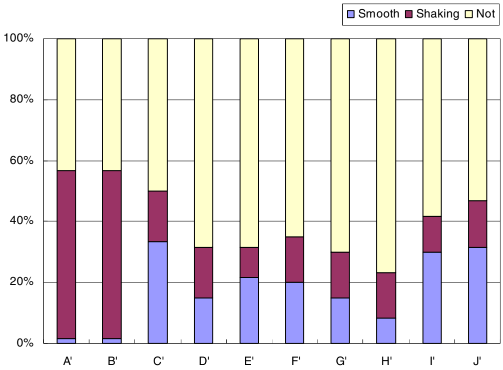


Fig. 8. Recognition rate for each pattern in set of medaka with insecticide

average is 40.5%. The shaking pattern has a higher observed rate than the smooth pattern regarding to the medaka treated with insecticide The lower recognition rate is observed in medaka that were treated with the insecticide compared to the medaka that were not treated with the insecticide. Unlike the data sets of specimens without the insecticide, the data sets A', B', C' which exceeded a 50% recognition rate showed many shaking patterns. However, the data sets of less than 50% showed a similar rate of smooth and shaking patterns.

5.2 Discussion

This study demonstrated that behavioral differences of animals in response to an insecticide could be detected by a decision tree using 3 features of behavior. One difficulty of conducting this type of monitoring study is the necessity of handling a large amount of data. In this study, the data were produced in every 0.25 second interval continuously for each individual measurement period. This produced a gigantic amount of data. The automatic pattern recognition system solved this time-consuming problem in detecting response behaviors. Besides time consumption in recognition, objectivity in judgments for classification has been another problem for manual recording. The application of machine intelligence to behavioral data has the advantage of classifying the movement patterns on a more objective basis. In this regard, the pattern recognition by a decision tree was demonstrated as an alternative to detecting the movement tracks of animals. These points will help in the analysis of behavior patterns in not only temperature elevation but also in different types of chemicals.

Another problem that arouses from this experiment is that biological specimens such as the medaka show too many different types of movement patterns. This makes selecting certain characteristics for a pattern difficult. This is why so many artificial systems such as neural networks and fuzzy are being used [31], [32], [33], [34]. However, although neural networks are sufficiently able to differentiate patterns, it is impossible to interpret exactly how much a certain pattern the specimen shows.

Results revealed that after differentiating smooth and shaking patterns through a decision tree, insecticide treatment caused the shaking ratio to increase. This can be seen as a pattern that appears in response to sub-lethal treatments of an insecticide, and is a process of adaptation. Smooth patterns show less angle change than shaking patterns and can be seen as a pattern without insecticide treatment, and it can be said that it appears the most frequently. Speed ratio of the medaka shows whether it is an active movement or inactive movement as shown in Fig. 6. Also, the area of projection product interprets smooth or shaking pattern. Power of FFT distinguishes specific patterns from unknown patterns.

Through this research, the decision tree logic was devised using 4 characteristic patterns and 'not determined' for the patterns that could not be defined, based on the knowledge of experts. The decision tree was able to differentiate the 4 patterns based on the observation of the three variables. However, more research must be done in order to define the patterns that were 'not determined.' Also, in order to better observe the many movement patterns of the medaka, more data sets should be examined and studied.

Biologically, results showed that variables such as smooth ratio vs. shaking ratio distinguished before and after insecticide treatment in Fig. 7 and 8. It can be inferred from these results that the activity did increase as the treatment began to rise. Although this is a short period of time it may be seen as a case of fast acclimation to the insecticide by the medaka.

6 Conclusions

The complex movement data were used to construct a decision tree with 3 features that could represent the movement tracks of medaka: speed ratio, power of FFT, and x- and y-axes projection product. As new input data were given to the decision logic, it was possible to recognize the change of pattern by examining the availability of insecticide. In these cases, a new analysis can be done to add new patterns and update the decision tree. The results of the decision tree revealed that whether the medaka was affected by insecticide or not, it interpreted speed, angle, area of projection to x- and y-axes using decision tree logic. If this is applied to more data sets, it is thought that more distinctive and accurate methods of differentiating the behavior patterns can be created. Also, this research in differentiating patterns may help in the field of research for the special characteristics of living organisms. This research can help the biosensor field in detecting defects in fish, or in finding out other chemical toxicants that exist in the water by observing specific behavior patterns of fish.

Acknowledgement

This work was supported by the grant No. "R01-2001-000-00087-0" from the Korea Science and Engineering Foundation.

References

1. Breiman, L., J. H. Friedman, R. A. Olshen, and C.G.Stone.: Classification and Regression Trees, Wadsworth International Group, Belmont, California. USA (1984)
2. Ripley, B. D.: Pattern recognition and neural networks. Cambridge University Press, Cambridge, UK (1996)
3. Richard, O. D., Peter, E.H., David, G.S.: Pattern Classification 2nd edn. Wiley Interscience, USA (2001)
4. Dutta, H., Marcelino, J., Richmonds, Ch.: Brain acetylcholinesterase activity and optomotor behavior in bluefills, Lepomis macrochirus exposed to different concentrations of diazinon. Arch. Intern. Physiol. Biochim. Biophys., 100(5) (1993) 331-334
5. Lemly, A. D., Smith, R. J.: A behavioral assay for assessing effects of pollutants of fish chemoreception. Ecotoxicology and Enviornmental Safety 11(2) (1986) 210-218
6. Roast, S. D., Widdows, J., Jones, M. B.: Disruption of swimming in the hyperbenthic mysid Neomysis integer (Peracarida: Mysidaces) by the organophosphate pesticide chlorpyrifos. Aquatic Toxicology 47 (2000) 227-241
7. Ibrahim, W. L. F., Furu, P., Ibrahim, A. M., Christensen, N. O.: Effect of the orgaophosphorous insecticide, chlorpyrifos (Dursban), on growth, fecundity and mortality of Biomphalaria aexandrina and on the production of Schistosoma mansoni cercariae in the snail, Journal of Helminthology 66 (1992) 79-88
8. Moore, A., Waring, C. P.: Sublethal effects of the pesticide diazinon on olfactory function in mature male Atlantic salmon parr, Journal of Fish Biology 48 (1996) 758-775
9. Gray, M. A., Teather, K. L., Metcalfe, C. D.: Reproductive success and behavior of Japanese medaka (*Oryzias latipes*) exposed to 4-tera-octylphenol, Environmental Toxicology and Chemistry 18 (1999) 2587-2594
10. Chon, T. S., Park, Y. S., Ross, M. H.: Activity of German cockroach, Blattella germanica (L.) (Orthoptera : Blattellidae), at different microhabitats in semi-natural conditions when treated with sublethal doses of pesticides, Journal of Asia-Pacific Entomology 1 (1998) 77-83
11. Bell, W. J.: Searching behavior patterns in insects. Annual Review of Entomology 35, (1990) 447-467
12. Alt, W., Hoffman, G. (Eds): Biological Motion. Lecture notes in Biomathematics 89. Springer-Verlag, Berlin (1989)
13. Tourtellot, M. K., Collins, R. D., Bell, W. J.: the problem of movelength and turn definition in analysis of orientation data. Journal of Theoretical Biology 150 (1991) 287-297
14. Bell, W. J., Kramer, E.: Search and anemotactic orientation of cockroach. Journal of Insect Physiology 25 (1975) 631-640
15. Bell, W. J., Kramer, E.: Sex pheromone-stimulated orientation of the American cockroach on a servosphere apparatus. Journal of Chemical Ecology 6 (1980) 287-295
16. Bell, W. J., Tobin, R. T.: Orientation to sex pheromone in the American cockroach: analysis of chemo-orientation mechanisms. Journal of Insect Physiology 27 (1981) 501-508

17. Sorensen, K. A., Bell, W. J.: Orientation responses of an isopod to temporal changes in relative humidity simulation of a "humid patch" in a "dry habitat," Journal of Insect Physiology 32 (1986) 51-57
18. White, J., Tobin, T. R., Bell, W. J., 1984. Local search in the house fly Musca domestica after feeding on sucrose. Journal of Insect Physiology 30 (1984) 477-487
19. Scharstein, H.: Paths of carabid beetle walking in the absence of orienting stimuli and the time structure of their motor output. In: Alt, W., Hoffmann, G. (Eds), Biological Motion. Lecture Notes in Biomathematics 89. Springer-Verlag, Berlin (1989) 269-277
20. Johnson, A. r., Milne, B. T., Wiens, J. A.: Diffusion in fractal landscapes: simulations and experimental studies of tenebrionid beetle movements. Ecology 73 (1992) 1968-1983
21. Weins, J. A., Crist, T. O., With, K. A., Milne, B. T.: Fractal patterns of insect movement in microlandscape mosaics. Ecology 79 (2) (1995) 663-666
22. Tchernichovski, O., Benjamini, Y., Golani, I.: The dynamics of long-term exploration in the rat Part I A phase-plane analysis of the relationship between location and velocity. Biological Cybernetics 78 (1998) 423-432
23. Tchernichovski, O., Benjamini, Y.: The dynamics of long-term exploration in the rat Part II A phase-plane analysis of the relationship between location and velocity. 78 (1998) 433-440
24. Quinlan, J. R.: Discovering rules by induction from large collections of examples. In: Micjie, E. (Ed.), Expert Systems in the Micro-Electornic Age, Edinburgh University Press, Edinburgh. (1979) 168-201
25. Quinlan, J. R.: Induction of decision trees. Machine Learning, 1(1) (1986) 81-106
26. Quinlan, J. R.: C4.5: programs for machine Learning. Morgan Kaufmann, San Francisco, CA, (1993)
27. Quinlan, J. R.: Improved use of continuous attributes in C4.5, Journal of Artificial Intelligence, 4. (1996) 77-90
28. Quinlan, J. R., Ronald L.: Rivest.Inferring decision trees using the minimum description length principle, Information and Computation. 80(3) (1989) 227-248
29. Tom M. Mitchell.: Machine Learning. McGraw-Hill, New York. (1997)
30. Kreyszig, Erwin: Advanced Engineering Mathematics, 8th Ed, Wiley. (1999)
31. Chon, T.-S., Park, Y. S., Moon, K. H., Cha, E. Y.: Patternizing communities by using an artificial neural network, Ecological Modeling 90 (1996) 69-78
32. I. S. Kwak, T. S. Chon, H. M. Kang, N. I. Chung, J. S. Kim, S. C. Koh, S. K. Lee, Y. S. Kim : Pattern recognition of the movement tracks of medaka (*Oryzias latipes*) in response to sub-lethal treatments of an insecticide by using artificial neural networks. Environmental Pollution, 120 (2002) 671-681
33. S. S. Kim, H. Bae, M. H. Lee: Design and optimization of fuzzy controllers based on the operator's knowledge, Artificial Life and Robotics 6 (2002) 92-98
34. Y. K. Woo, H. Bae, S. S. Kim, K. B. Woo: Intelligent Methods to Extract Knowledge from Process Data in the Industrial Applications. International Journal of Fuzzy Logic and Intelligent Systems, 3(2) (2003) 194-199

Support Vector Inductive Logic Programming

Stephen Muggleton[1], Huma Lodhi[1],
Ata Amini[2], and Michael J. E. Sternberg[2]

[1] Department of Computing, Imperial College,
180 Queen's Gate, London, SW7 2AZ, UK
{shm, hml}@doc.ic.ac.uk

[2] Department of Biological Sciences, Imperial College,
180 Queen's Gate, London, SW7 2AZ, UK
{ata.amini, m.sternberg}@imperial.ac.uk

Abstract. In this paper we explore a topic which is at the intersection of two areas of Machine Learning: namely Support Vector Machines (SVMs) and Inductive Logic Programming (ILP). We propose a general method for constructing kernels for Support Vector Inductive Logic Programming (SVILP). The kernel not only captures the semantic and syntactic relational information contained in the data but also provides the flexibility of using arbitrary forms of structured and non-structured data coded in a relational way. While specialised kernels have been developed for strings, trees and graphs our approach uses declarative background knowledge to provide the learning bias. The use of explicitly encoded background knowledge distinguishes SVILP from existing relational kernels which in ILP-terms work purely at the atomic generalisation level. The SVILP approach is a form of generalisation relative to background knowledge, though the final combining function for the ILP-learned clauses is an SVM rather than a logical conjunction. We evaluate SVILP empirically against related approaches, including an industry-standard toxin predictor called TOPKAT. Evaluation is conducted on a new broad-ranging toxicity dataset (DSSTox). The experimental results demonstrate that our approach significantly outperforms all other approaches in the study.

1 Introduction

In this paper we propose a novel machine learning approach which combines the dimensionality independence advantages of Support Vector Machines (SVMs) with the expressive power and flexibility of Inductive Logic Programming (ILP). In particular, we propose a kernel which is an inner product in the feature space spanned by a given set of first order hypothesised clauses. As with normal ILP, examples, background knowledge and hypothesised clauses are encoded as logic programs. The kernel not only captures the semantic and syntactic relational information contained in the data but also provides the flexibility of using arbitrary forms of structured and non-structured data.

The approach we suggest differs from the relational kernels suggested in [1,2] by our use of logical background knowledge. In order to understand the distinction being made here consider the following three settings for ILP.

Atomic Generalisation. This setting is characterised by having examples of which are typically ground atomic formulae and hypotheses consisting of atomic formulae which

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 163–175, 2005.

entail the examples. Plotkin [3] showed that this hypothesis space forms a lattice which is partially ordered by atomic subsumption.

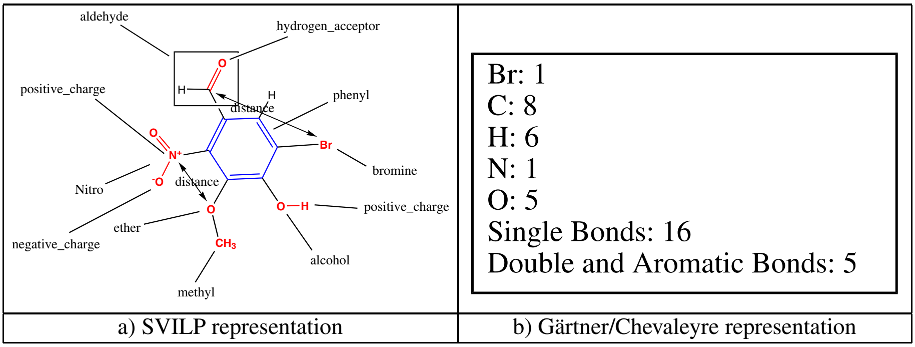
Clausal Generalisation. In this setting examples are ground clauses and hypotheses are clauses which entail the examples. Plotkin [4] showed that once more this hypothesis space forms a lattice which is partially ordered by clausal subsumption.

Clausal Generalisation Relative to Background Knowledge. This third setting [4] is distinguished by assuming the existence of background knowledge in the form of a conjunction of clauses. Examples are ground clauses. Hypotheses are clauses which when conjoined with the background knowledge entail the examples.

Most ILP research has assumed the third setting, clausal generalisation relative to background knowledge, since this is the more general approach. The use of background knowledge provides a flexible way of encoding the understanding of domain experts, and can increase both the predictive accuracy of the learning and the degree of insight provided relative to the background knowledge. However, this setting brings with it overheads related to the theorem proving involved in using background knowledge. For this reason Page and Frisch [5] investigated the use of atomic generalisation with respect to a monadic constraint theory. This is a generalisation of the first setting, and a special case of the third setting.

More recently Lloyd [6] and others [2] have investigated algorithms which use the setting of atomic generalisation, but with more general forms of strongly-typed terms. In particular, terms can consist of arbitrary sets. This allows more flexibility for defining data types without the overheads associated with background knowledge. In [2] it is shown that this form of representation and learning can be used to formulate a relational kernel. In [1] it is shown that by using the “bag of atoms” representation introduced in [7] a multi-instance kernel approach can even be applied to structurally complex ILP learning problems involving small molecules.

The SVILP approach is a form of generalisation relative to background knowledge, though the final combining function for the ILP-learned clauses is an SVM rather than a logical conjunction. We will now provide a simplified worked example to show the difference in representing molecules using the Gärtner/Chevaleyre approach from the representation used by our SVILP kernel. Figure 1 shows a typical molecule from the DSSTox dataset of toxins (see Section 5). In the SVILP approach we start by formulating chemical background knowledge in the form of Prolog definitions. These have been designed by one of the authors (Ata Amini), a biochemistry domain expert, to be relevant to properties associated with toxins. Such properties include the existence of substructures such as aromatic rings, methyl and alcohol substructures, types of atom, charge, the existence of hydrogen acceptors and distances between various critical structures on the molecule. The ILP system CProgol5.0 is used to generate a set of hypothesised clauses based on the given background knowledge and examples. An SVM kernel is then used as the combining function for predictions of these clauses. By contrast, the Gärtner/Chevaleyre features consist simply of the frequency of occurrence of atoms, bonds and atom pairs within the given molecule. These are used to form a vector representation of the molecule. An obvious advantage of this “bag-of-atoms” representation is that it requires no domain expertise and thus is less effort to develop. By analogy with



a) SVILP representation

b) Gärtner/Chevaleyre representation

Fig. 1. Molecule represented using a) SVILP representation which employs a kernel based on domain-expert informed chemical background knowledge indicated by the annotations on the figure and b) Gärtner/Chevaleyre bag-of-atoms uses Multi-Instance (MI) kernel based on frequency of occurrences of atoms and atom pairs

the use of the “bag-of-words” [8] representation in text classification one might expect a simple representation of this form to lead to superior predictive accuracy. However, this is not the case in the experiments reported in Section 5 in which the SVILP kernel significantly outperforms the Gärtner/Chevaleyre kernel. In this case, the use of more highly informed background knowledge in the SVILP appears to provide a significant advantage.

The paper is arranged as follows. The Background Section 2 introduces the basic ideas behind kernels, SVMs and Inductive Logic Programming (ILP). In Section 3 SVILPs are defined and their properties proved. This is followed by a section which describes Related Work (Section 4). Next we describe the Experiments (Section 5) on toxicity data. The paper then concludes.

2 Background

Kernels and Support Vector Machines: During recent years, there has been increasing interest in kernel-based methods such as Support Vector Machines (SVMs) [9]. The non-dependence of these methods on the dimensionality of the feature space and the flexibility of using any kernel make them a good choice for different tasks such as classification and regression. We can view the learning process of SVMs as comprising two stages. 1) Map the input data, $d_1, \ldots, d_n \in D$, into some higher dimensional space H through a non-linear mapping ϕ that is given by $\phi : D \rightarrow H$. The mapping ϕ may not be known explicitly but be accessed via the kernel function described below. 2) Construct a linear function f in the space.

The kernel function $K(d_i, d_j) = \langle \phi(d_i), \phi(d_j) \rangle$ computes the inner product between the mapped instances. The mathematical foundation of such a function was established during the first decade of the twentieth century [10]. A kernel function is a symmetric function, $K(d_i, d_j) = K(d_j, d_i) \quad \text{for } i, j = 1, \ldots, n$, and satisfies the property of positive semi-definiteness, $\sum_{1,j=1}^{n} a_i a_j K(d_i, d_j) \geq 0 \quad \text{for } a_i, a_j \in R$.

The $n \times n$ matrix with entries of the form $K_{ij} = K(d_i, d_j)$ is known as the kernel matrix or the Gram matrix. A kernel matrix is a symmetric, positive definite matrix. In other words the n Eigen values of this $n \times n$ kernel matrix are non-negative. Kernel functions can be defined over general sets [11]. This important fact has allowed successful exploration of novel kernels for discrete spaces such as strings and graphs [12,13].

Inductive Logic Programming: Inductive Logic Programming (ILP) [14] is the area of AI which deals with the induction of hypothesised predicate definitions. In ILP logic programs are used as a single representation for examples, background knowledge and hypotheses. ILP is differentiated from most other forms of Machine Learning (ML) both by its use of an expressive representation language and its ability to make use of logically encoded background knowledge. This has allowed successful applications of ILP in areas such as molecular biology [15] and chemoinformatics [16].

In the following it is assumed that the examples, background knowledge and hypotheses each consist of logic programs, ie sets of first-order Horn clauses. The normal semantics of ILP is as follows. We are given background (prior) knowledge B and evidence E. The evidence $E = E^+ \wedge E^-$ consists of positive evidence E^+ and negative evidence E^-. The aim is then to find a hypothesis H such that the following conditions hold.

Prior Satisfiability. $B \wedge E^- \not\models$
Posterior Satisfiability. $B \wedge H \wedge E^- \not\models$
Prior Necessity. $B \not\models E^+$
Posterior Sufficiency. $B \wedge H \models E^+$

Since a large number of hypotheses will typically fit such a definition, the Bayesian ILP setting [17] assumes a prior probability distribution defined over the hypothesis space. Algorithms such as CProgol [18] use such a prior to search for hypotheses which maximise the posterior probability $p(H|E)$.

3 Support Vector Inductive Logic Programming (SVILP)

The SVILP framework builds on the ILP framework. Thus we also assume background knowledge B, examples E and a hypothesis H for which the conditions of the normal semantics hold. The key difference between ILP and SVILP is the way in which the set of clauses H is used for predictive purposes. In ILP H is simply treated as a conjunction, for which any instance d from the domain of instances D is predicted to be true if and only if $B, H \models d$.

By contrast, SVILP bases a kernel on the predictions of the clauses h in H. This involves forming a binary hypothesis-instance association matrix M in which element $M_{ij} = 1$ (0 otherwise) if and only if clause $h_i \in H$ entails instance $d_j \in D$ as follows, $B, h_i \models d_j$.

The kernel described in Section 3.2 can be viewed as a function for which similarity of two instances d_1 and d_2 is based on the similarity of the rows of clauses in M associated with d_1 and d_2.

father(henry,john). father(david,henry). mother(jane,john). mother(elizabeth,henry).
father(charles,mary). father(egbert,jill). mother(jill,mary). mother(ann,jill).

grandfather(F,P) ← father(F,P1), parent(P1,P).
grandmother(M,P) ← mother(M,P1), parent(P1,P).
parent(F,P) ← father(F,P).
parent(M,P) ← mother(M,P).

hair(john,blond). hair(mary,black).
hair(henry,blond). hair(charles,black).
hair(jill,blond). hair(elizabeth,blond).
hair(egbert,black). hair(ann,blond).
hair(david,black). hair(jane,black).

Fig. 2. Background knowledge for disease inheritance

3.1 Family Example

In this artificial example we assume that the occurrence of a disease is related to the inheritance patterns of an observable property (e.g., hair colour) in various families. The background knowledge is shown in Figure 2. This describes the relationships in the family tree shown in Figure 6. Examples of individuals having the disease are shown in Figure 3 and various hypothesised clauses are shown in Figure 4. Assuming the domain is limited to the examples, we show the resulting binary hypothesis-instance association matrix in Figure 5.

1. disease(john) 2. disease(mary) 3. disease(jane) 4. disease(henry) 5. disease(charles)

Fig. 3. Examples for disease inheritance

A. disease(P) ← hair(P,Colour), father(F,P), hair(F,Colour)
B. disease(P) ← hair(P,Colour), mother(M,P), hair(M,Colour)
C. disease(P) ← hair(P,Colour), grandmother(M,P), hair(M,Colour)
D. disease(P) ← hair(P,Colour), grandfather(F,P), hair(F,Colour)
E. disease(P) ← hair(P,black), father(F,P), mother(M,P),
hair(F,blond), hair(M,black)
F. disease(P) ← hair(P,black), father(F,P), grandfather(G,P),
hair(F,blond), hair(G,black)

Fig. 4. Hypothesised clauses for disease inheritance

	A	B	C	D	E	F
1	1	0	0	1	0	0
2	1	0	0	1	0	0
3	1	0	0	0	0	0
4	0	1	0	0	0	0
5	1	0	0	0	0	0

Fig. 5. Resulting binary hypothesis-instance association matrix

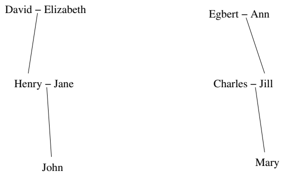


Fig. 6. Family trees for disease inheritance

Note that according to the matrix examples 1 and 2 have maximum similarity. This is despite the fact that the hair colour (the main observable feature) of John and Mary (the individuals involved in the examples) are opposite (blond and black respectively). The example demonstrates the strong learning bias which can be introduced by the use of background knowledge and hypotheses within the SVILP setting. In the next section we define the kernel formally.

3.2 Definition of Kernel

We assume background knowledge B and a set of hypothesised clauses H drawn from a class of hypotheses $\mathcal{H}$ and a set of instances D drawn from a class of instances $\mathcal{D}$. Each hypothesis clause h in H can be thought of as a function of the following form, $h : \mathcal{D} \rightarrow \{\text{True}, \text{False}\}$. Conversely the τ function gives the hypothesised clauses covering any particular instance, $\tau : \mathcal{D} \rightarrow 2^H$. Where for any d_i in D

$$\tau(d_i) = \{h : \exists h \in H, (B, h \models d_i)\}$$

As in the Bayesian ILP framework [17], we assume a prior probability distribution over the hypotheses. This can be represented as a function π such that

$$\pi : H \rightarrow [0, 1] \quad \text{and} \quad \sum_{h \in H} \pi(h) = 1$$

Next we define a function, which maps sets of hypothesised clauses to probabilities.

$$f : 2^H \rightarrow [0, 1]$$

For all $H' \subseteq H$

$$f(H') = \sum_{h \in H'} \pi(h)$$

Now the kernel function is as follows. For all d_i, d_j in D

$$K(d_i, d_j) = f(\tau(d_i) \cap \tau(d_j))$$

It can be easily shown that the kernel is an inner product in ILP space.The kernel requires a hypothesised clause set H. In order to improve the informative power of the kernel we define a prior probability distribution and fits the prior to the coordinates in space spanned by the hypothesised clauses. In this way a countable set of hypothesised clauses implies a mapping ϕ that maps the data into an ILP space, where dimensionality of the space is the same as the cardinality of the set of hypothesised clauses and each mapped instance can have r number of non-zero entries (in a column vector) where r is in the range $1 \leq r \leq k$. Formally

$$f_i(d) = \sqrt{\pi(h_i(d))} \quad \text{for } i = 1, \ldots, k$$

Hence the mapping ϕ for an instance is given by

$$\phi : d \rightarrow ((f_1(d), f_2(d), \ldots, f_k(d)) = (f_i(d)_{i=1}^k)$$

and kernel for instance d_i and d_j is given by

$$K(d_i, d_j) = \langle \phi(d_i), \phi(d_j) \rangle = \sum_{i=1}^{k} f_i(d_i) f_i(d_j)$$

Hence, $$K(d_i, d_j) = f(\tau(d_i) \cap \tau(d_j))$$

The validity of kernel function follows from the definition as an inner product however we can show that it satisfies Mercer's condition (symmetry and positive semi-definiteness). Clearly the kernel function is symmetric and positive semi-definiteness occurs since there is mapping ϕ from D into an ILP space. For all $a_i \in R$ and $d_i \in D$, for $i = 1, \ldots, n$ we have the following expression, $\sum_{1,j=1}^{n} a_i a_j K(d_i, d_j)$. We now use a compact representation $A = (a_i)_{i=1}^{n}$ and $\phi = (\phi(d_i))_{i=1}^{n}$, hence kernel matrix $\sum_{i,j=1}^{n} K(d_i, d_j) = \phi\phi'$ and the expression is, $A\phi'\phi A' = t't \geq 0$.

Given that ϕ maps the data into ILP space, we can construct Gaussian RBF kernels in ILP space $K_{RBF}(d_i, d_j) = \exp\left(\frac{-\|(\phi(d_i) - \phi(d_j)\|^2}{2\sigma^2}\right)$, where $\|(\phi(d_i) - \phi(d_j)\| = \sqrt{K(d_i, d_i) - 2K(d_i, d_j) + K(d_j, d_j)}$. Our method is flexible to construct any kernel in the space spanned by the clauses. However we select RBF kernels (K_{RBF}) constructed in ILP space for our experiments in section 5.

We now consider the analysis of the complexity of the kernel. Assuming the theorem prover can test each hypothesised clause against each instance in time bounded by a constant k, the overall time taken to compute the kernel is proportional to the number of hypothesised clauses $|H|$ and the number of instances $|D|$.

4 Related Work

Propositionalisation: Within ILP "propositionalisation" [19] techniques transform machine learning problems from a first-order logic setting into one which can be handled by a "propositional" or feature-based learner. Kramer et al. [19] distinguish between domain-independent ([20,21,22,23]) and domain-dependent approaches (eg [24]). In most domain-independent propositionalisation approaches [21,22,23] features are introduced as clauses with a monadic head predicate. For instance, when applied to problems involving molecular descriptions these techniques introduce new features such as the following.

```
f1(A):-has_rings(A, [R1, R2]), hydrophobic(A,H), H > 1.0.
f2(A):-atm(A,B,_,27,_), bond(A,B,C,_), atm(A,C,_,29,_).
```

Though superficially similar to domain-independent propositionalisation, the SVILP approach described in this paper is not a propositionalisation technique since it does not transform the representation by the introduction of such monadic features. Instead a general-purpose ILP learning algorithm is used to learn clauses with heads having arbitrary predicate arities. The heads of these clauses can contain terms with multi-arity function symbols and constants. In normal ILP the hypothesis used for predictive purposes would consist of these clauses conjoined together. In SVILP the truth-value predictions of these individual clauses are projected onto the instance space. The kernel matrix is then formulated over the instance-space predictions of the individual clauses.

SVILP is similar in its use of support-vector technology to the domain-dependent propositionalisation approach of Kramer and Frank [24]. This uses bottom-up evaluation to fine. The key difference here is that SVILP is domain-independent, allowing the use of background knowledge to encode the appropriate machine learning bias.

Kernels within ILP: Within ILP there has recently been interest in the development of kernels which incorporate relational information, for use within support vector machines[2,25,26,27]. Several authors [2,25] take the approach of using syntactic measures of distance between first-order formulae as the basis for such kernels. Within the ILP literature it is normal to differentiate between *syntactic* [28,29] and *semantic* [30] distance measures. Syntactic measures are based on differences in the structure of first-order formulae, and tend to be confined to comparison of terms, rather than arbitrary first-order formulae. Semantic measures are based on comparison of models, making this approach intractable for all but simple formulae.

The kernel approaches described in [2,25] are unable to make use of background knowledge, since they are based on syntactic comparison of ground atoms. By contrast, a central feature of the SVILP described in this paper is its use of generalisation relative to background knowledge.

5 Experiments

A new dataset was used for evaluating SVILP. The DSSTox dataset was made available to us by Dr Ann Richards of National Health and Environmental Effects Research Laboratory, USA. The dataset represents the most diverse set of toxins presently available in the public domain. By choosing a new toxin dataset we avoided over-testing problems associated with molecular datasets such as the Mutagens [31]. The 188 molecule Mutagenic dataset has now been evaluated by so many researchers that it is becoming hard to argue that some of the higher reported accuracies are not simply due to chance.

Permethrin
Flucythrinate

Fig. 7. Examples of compounds in DSSTox

Materials: The DSSTox [32] database contains organic and organometalic molecules with their toxicity values. The dataset consists of 576 molecules. Figure 7 shows an example of two of the molecules found in DSSTox. As far as we know no previous attempt has been made to quantify the structure and activity relationship for the whole DSSTox dataset.

Methods: We now describe the pre-processing stage. Molecules in the form of SMILES strings, were transformed into 3D structures using the software CONCORD 4.0 [33] (implemented in TRIPOS). All of the molecules contain continuous chemical feature known as the lowest unoccupied molecule orbital (LUMO), water/octanol partition coefficient (LOGP) and dipole moment. LOGP reflects the hydrophobicity of compounds and the mechanism of toxicities of these chemicals are based on their accumulation in the non-polar lipid phase of the biomembranes. LUMO and dipole moment can describe electrophilicities of compounds. The key information is given in the form of atom and bond description.

We compared the performance of SVILP with a number of related techniques including partial least squares (PLS), multi instance kernels (MIK) [1,2], an RBF kernel using only 3 chemical features (LOGP, LUMO, dipole moment) that we term as CHEM. We also compared the performance of SVILP with well known QSAR software TOPKAT (Toxicity Prediction by Komputer Assisted Technology).

As our experimental methodology we used 5-fold cross validation. For evaluation we used mean squared error (MSE) and R-squared (standard measure of accuracy in QSAR). In this work we employed $\epsilon-$insensitive SVM regression (SVR)[9]. We used the SVM package SVMTorch [34] for our experiments. C (regularization parameter), ϵ (controls width of insensitive band), σ (width of Gaussian) are the tunable parameters for kernel-based methods (SVILP, CHEM and MIK). In PLS the tunable parameter is the "number of components". These parameters can be set by some model selection method. The traditional protocol to set the values for the parameters is the minimisation (maximisation) of some criterion relative to the values of the parameters using a validation set. We select the optimal values of the tunable parameters using a validation set as described. We set the parameters for each fold using only the training set of the fold. We randomly selected a subset comprising 75% of the data (training set of each fold) for the training set and used the remaining data as a test set. A range of values of the parameters were selected. The sets of the values are given by $C = \{10, 100, 1000, 10000\}$, $\epsilon = \{0.1, 0.3, 0.5, 1.0\}$, $\sigma = \{0.125, 0.25, 0.5, 4, 16\}$. For PLS we used the number of components from 1 to 15. The parameters which give the minimum MSE on the validation set were chosen. For the selected parameters we obtained the models (created by the methods) using full training set and performed evaluation on test compounds.

In order to perform the prediction task using SVILP, we first obtained a set of clauses. Examples and Background knowledge (atom-bond, high level chemical groups e.g. phenyl ring, aldehyde, carboxylic acids and chemical features) are given to CProgol5.0 [18] which generates a set of hypothesised clauses. For all the folds, the clauses with positive compression were selected where the number of obtained clauses for each fold can vary between 1500-2000. The compression value of a clause is given by $V = \frac{P*(p-(n+c+h))}{p}$, where p is the number of positive instances correctly deducible from the clause, n is the number of negative examples incorrectly deducible from the clause, c is the length of the clause and h is number of further atoms to complete the input/output connectivity of the clause and P is the total number of positive examples. The hypothesised clauses are then taken by a Prolog program which computes the hypothesis-instance association (see Section 3), indicating for each instance the set of all hypothesised clauses which imply it. In this work we used a uniform probability

	MSE	R-squared
CHEM	0.811	0.519
PLS	0.671	0.593
MIK	0.838	0.503
SVILP	**0.574**	**0.655**

Fig. 8. MSE and R-squared for CHEM, PLS, MIK and SVILP

	Accuracy
ILP (CProgol5.0)	55
CHEM	58
PLS	71
MIK	60
SVILP	**73**

Fig. 9. Accuracy for ILP, CHEM, PLS, MIK and SVILP

distribution over the clauses. We then computed the similarity between molecules using proposed kernel. In order to apply PLS for toxicity prediction, we used the same set of hypothesised clauses generated by CProgol5.0 as SVILP.

Results: We conducted a series of experiments to evaluate the performance of the proposed method. We conducted the first set of experiments to evaluate the efficacy of the new method for predicting the toxicity values. Figure 8 shows the results. The results are averaged over 5 runs of the methods. Based on the statistical sign test method, SVILP shows significant improvement in comparison with the other methods in the study. In the second set of experiments we assessed the performance of the methods for qualitative prediction. We evaluated our approach by employing it for categorising the molecules into two categories, toxic and non-toxic. We also compared the performance of SVILP with the standard ILP system CProgol5.0. All of the methods predict the non-toxic molecules with high accuracy. Figure 9 shows the results for the category ”toxic”. According to McNemar test the SVILP method shows significant improvement with respect to the other methods. We finally compared SVILP with TOPKAT. The software accepts the structures of the molecules in SMILES string and automatically split the molecule into different fragments, and then uses these fragments as well as some chemical descriptors such as LOGP and shape index for predictions. In order to make

	MSE	R-squared
CHEM	1.04	0.48
PLS	1.03	0.47
TOPKAT	2.2	0.26
SVILP	**0.8**	**0.57**

Fig. 10. MSE and R-squared for CHEM, PLS, TOPKAT and SVILP

a fair comparison of the above methods with the commercial software TOPKAT, we must ensure that we only consider predicted accuracies for molecules that were not included in the training data of either method. We therefore excluded any of the DSSTox molecules that TOPKAT had in its database leaving 165 unseen molecules. Figure 10 shows the results. According to sign test, the SVILP shows significant improvement in comparison with all of the other approaches. Our results show that SVILP outperforms all the other methods in the study. The results confirm the efficacy and usefulness of our approach.

6 Conclusions and Further Work

In this paper we introduce a new framework for combining Support Vector machine technology with Inductive Logic Programming. Unlike existing relational kernels, the present approach works within the standard ILP setting of generalisation with respect to background knowledge, rather than the limited setting of atomic generalisation. A particular kernel is defined and implemented on top of the ILP system CProgol5.0. This kernel has been tested on an important new toxin dataset. In our experiments we compared the performance of the SVILP against related approaches. In all cases our approach produced significantly higher predictive accuracy.

Further theoretical work is necessary to clarify the effects on performance of varying the amount of background knowledge used by the kernel. Also further empirical work is needed to test the kernel on a wider variety of relational problems.

Acknowledgements

The authors would like to acknowledge the support of the DTI Beacon project "Metalog - Integrated Machine Learning of Metabolic Networks Applied to Predictive Toxicology", Grant Reference QCBB/C/012/00003 and the ESPRIT IST project "Application of Probabilistic Inductive Logic Programming II (APRIL II)", GrantRef: FP-508861.

References

1. Gärtner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In: Proceedings of the Nineteenth International Conference on Machine Learning. (2002) 176–186
2. Gärtner, T., Lloyd, J.W., Flach, P.A.: Kernels for structured data. In Matwin, S., Sammut, C., eds.: Proceedings of the Twelfth International Conference on Inductive Logic Programming. LNAI 2583, Berlin, Springer-Verlag (2002) 66–83
3. Plotkin, G.: A note on inductive generalisation. In Meltzer, B., Michie, D., eds.: Machine Intelligence 5. Edinburgh University Press, Edinburgh (1969) 153–163
4. Plotkin, G.: Automatic Methods of Inductive Inference. PhD thesis, Edinburgh University (1971)
5. Page, D., Frisch, A.: Generalization and learnability: A study of constrained atoms. In Muggleton, S., ed.: Inductive Logic Programming. Academic Press, London (1992)
6. Lloyd, J.: Logic for Learning. Springer, Berlin (2003)
7. Chevaleyre, Y., Zucker, J.: A framework for learning rules from multiple instance data. In: Proceedings of the European Conference on Machine Learning (ECML 2001), Berlin, Springer-Verlag (2001) 49–60 LNAI 2167.

8. Dumais, S., Platt, J., Heckermann, D., Sahami, M.: Inductive learning algorithms and representations for text categorisation. In: Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management. (1998) 148–155
9. Vapnik, V.: The Nature of Statistical Learning Theory. Springer Verlag, New York (1995)
10. Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. Philosophical Transactions of the Royal Society London (A) **209** (1909) 415–446
11. Haussler, D.: Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California in Santa Cruz, Computer Science Department (1999)
12. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. Journal of Machine Learning Research **2** (2002) 419–444
13. Horváth, T., Gaertner, T., Wrobel, S.: Cyclic pattern kernels for predictive graph mining. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2004) 158–167
14. Muggleton, S.: Inductive Logic Programming. New Generation Computing **8** (1991) 295–318
15. King, R., Whelan, K., Jones, F., Reiser, P., Bryant, C., Muggleton, S., Kell, D., Oliver, S.: Functional genomic hypothesis generation and experimentation by a robot scientist. Nature **427** (2004) 247–252
16. Sternberg, M., Muggleton, S.: Structure activity relationships (SAR) and pharmacophore discovery using inductive logic programming (ILP). QSAR and Combinatorial Science **22** (2003)
17. Muggleton, S.: Bayesian Inductive Logic Programming. In Warmuth, M., ed.: Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory, New York, ACM Press (1994) 3–11 Keynote presentation.
18. Muggleton, S.: Inverse entailment and Progol. New Generation Computing **13** (1995) 245–286
19. Kramer, S., Lavrac, N., Flach, P.: Propositionalisation approaches to Relational Data Mining. In Dzeroski, S., Larac, N., eds.: Relational Data Mining. Springer, Berlin (2001) 262–291
20. Lavrač, N., Džeroski, S., Grobelnik, M.: Learning non-recursive definitions of relations with LINUS. In Kodratoff, Y., ed.: Proceedings of the 5th European Working Session on Learning, volume 482 of Lecture Notes in Artificial Intelligence, Springer-Verlag (1991)
21. Kramer, S., Pfahringer, B., Helma, C.: Stochastic propositionalisation of non-determinate background knowledge. In: Proceedings of the Eighth International Conference on Inductive Logic Programming, Berlin, Springer-Verlag (1998) 80–94
22. Srinivasan, A., King, R.: Feature construction with inductive logic programming: a study of quantitative predictions of biological activity aided by structural attributes. Data Mining and Knowledge Discovery **3** (1999) 35–57
23. Dehaspe, L., Toivonen, H.: Discovery of frequent datalog patterns. Data Mining and Knowledge Discovery **3** (1999) 7–36
24. Kramer, S., E, F.: Bottom-up propositionalisation. In: Proceedings of the ILP-2000 Work-In-Progress Track. Imperial College, London (2000) 156–162
25. Mavroeidis, D., Flach, P.: Improved distances for structured data. In Horváth, T., Yamamoto, A., eds.: Proceedings of the Thirteenth International Conference on Inductive Logic Programming. LNAI 2835, Berlin, Springer-Verlag (2003) 251–268
26. Cumby, C., Roth, D.: On kernel methods for relational learning. In: Proceedings of the Twentieth International Conference on Machine Learning. (2003) 107–114
27. Gaertner, T., Driessens, K., Ramon, J.: Graph kernels and gaussian processes for relational reinforement learning. In: Proc. of the 13th International Conference on Inductive Logic Programming, Springer Verlag (2003) 146–163

28. Ramon, J., Bruynooghe, M.: A framework for defining distances between first-order logic objects. In Page, D., ed.: Proceedings of the Eighth International Workshop on Inductive Logic Programming (ILP98), Berlin, Springer-Verlag (1998) 271–280 LNAI 1446.
29. Horvath, T., Wrobel, S., Bohnebeck, U.: Relational instance-based learning with lists and terms. Machine Learning **43** (2001) 53–80
30. Nienhuys-Cheng, S.: Distance between Herbrand interpretations: a measure for approximations to a target concept. In Lavrač, N., Džeroski, S., eds.: Proceedings of the Seventh International Workshop on Inductive Logic Programming (ILP97), Berlin, Springer-Verlag (1997) 321–226 LNAI 1297.
31. King, R., Muggleton, S., Srinivasan, A., Sternberg, M.: Structure-activity relationships derived by machine learning: the use of atoms and their bond connectives to predict mutagenicity by inductive logic programming. Proceedings of the National Academy of Sciences **93** (1996) 438–442
32. Richard, A., Williams, C.: Distributed structure-searchable toxicity (DSSTox) public database network: A proposal. Mutation Research **499** (2000) 27–52
33. Pearlman, R.S.: Concord User's Manual. Tripos, Inc, St Louis, Missouri (2000)
34. Collobert, R., Bengio, S.: Svmtorch: Support vector machines for large-scale regression problems. Journal of Machine Learning Research **1** (2001) 143–160

Measuring Over-Generalization in the Minimal Multiple Generalizations of Biosequences

Yen Kaow Ng[1], Hirotaka Ono[2], and Takeshi Shinohara[3]

[1] Kyushu Institute of Technology,
Graduate School of Computer Science and Systems, Iizuka, 820, Japan
kalngyk@daisy.ai.kyutech.ac.jp

[2] Kyushu University, Department of Computer Science and Communication Engineering, 6-10-1, Hakozaki, Fukuoka, 812-8581, Japan
ono@csce.kyushu-u.ac.jp

[3] Kyushu Institute of Technology,
Department of Artificial Intelligence, Iizuka, 820, Japan
shino@ai.kyutech.ac.jp

Abstract. We consider the problem of finding a set of patterns that best characterizes a set of strings. To this end, Arimura *et. al.* [3] considered the use of minimal multiple generalizations (mmg) for such characterizations. Given any sample set, the mmgs are, roughly speaking, the most (syntactically) specific set of languages containing the sample within a given class of languages. Takae *et. al.* [17] found the mmgs of the class of pattern languages [1] which includes so-called sort symbols to be fairly accurate as predictors for signal peptides. We first reproduce their results using updated data. Then, by using a measure for estimating the level of over-generalizations made by the mmgs, we show results that explain the high level of accuracies resulting from the use of sort symbols, and discuss how better results can be obtained. The measure that we suggests here can also be applied to other types of patterns, e.g. the PROSITE patterns [4].

1 Introduction

Finding patterns that characterize a set of samples is a common task in molecular biology [6]. The samples are genes that are known to share certain traits, while the discovered patterns are to be used for the study of the genes, or for predicting if an unknown gene will demonstrate similar traits.

Computational means to find such patterns vary from statistical approaches to methods used in language learning [6,7]. In language learning, the class of patterns most commonly used is the one which defines the *pattern languages* [1], in which case, a pattern is taken to be a string over a finite alphabet Σ and an infinite set $\{x_1, x_2, \ldots\}$ of variables, while the language generated by such a pattern is taken to be the strings obtainable by replacing all the variables in the pattern with strings over Σ. As an example, let $A, C \in \Sigma$, then the language of the pattern "Ax_1C" are the strings over Σ that begins with 'A' and ends with 'C'.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 176–188, 2005.

For such pattern languages, the problem of finding a characterization for a set of samples is frequently restated as the problem of looking for a language (or the unions of a set of languages) within a class that contains all the strings in the sample. Such approaches have been very actively studied within the learning theory community in the last decade, using either the pattern languages or its subclasses [14,8]. We consider the problem first introduced by Arimura *et. al.* [3], where we are to find a collection of up to k languages within a class $\mathcal{L}$ of pattern languages which (1) together contains all the elements in the given sample S, and (2) is the most "*specific*" set of languages (we formally define this notion of "most specific languages" in Section 2) among every union of up to k languages in the class that contains S. Hence each number k, class $\mathcal{L}$ of languages, and set S of samples completely specify an instance of such a problem. A set of patterns that fulfills these conditions is called a k-mmg. Note that more than one k-mmg may exist for any given problem instance. Arimura *et. al.* gave a generic polynomial time algorithm (**MMG**) [3] for finding a k-mmg, for the problem instances where certain conditions are fulfilled [18].

There are a number of studies that followed up on the **MMG** algorithm [2,19,16], using the class of *regular patterns* [15] as search space. The regular patterns are patterns where each variable may appear at most once in it. For example, "Ax_1Cx_2" has the variables x_1 and x_2 each appearing only once in it. (As a counter-example, x_1 appears twice in "Ax_1Cx_1".) Takae *et. al.* [17] added an element called *sort symbol* to the regular patterns, resulting in a new pattern class called *sort regular patterns.*

A sort symbol is a letter associated with a non-empty set $S \subseteq \Sigma$, and is written $[A_1A_2\ldots]$ where A_1, A_2, $\ldots$ is an enumeration of the elements in S. A sort symbol associated with a set S in a pattern p can be replaced with any element in S in generating elements of the pattern language of p. Such sort symbols have been in common use within the molecular biology community for some time [4], usually under the name of *ambiguous letter* or *ambiguous character.* Takae *et. al.* found that the k-mmgs of sort regular pattern languages achieve higher accuracies than those of regular pattern languages of the same k in the prediction of signal peptide functions in unknown biosequences.

The present work aims to clarify Takae *et. al.*'s observations.

We first note that any sort regular pattern can be expressed as the union of a few regular patterns. For example, "$x_1[AC]x_2$" is the union of "x_1Ax_2" and "x_1Cx_2". Hence allowing the use of sort symbols achieves similar effects as allowing more languages to be in a union. From this observation, we can perhaps compare the k-mmgs of sort regular patterns to the k'-mmgs of regular patterns for some k' that is "suitably" larger than k. However, such k' and k cannot be straight-forwardly decided from the alphabet size, or the number of occurrences of sort symbols that appeared in the k-mmg. This is because allowing for more languages to be union-ed provides more flexibility than simply allowing the use of sort symbols, since patterns in a union can bear no similarity at all to each other. In a more delicate example, we see that the union of the languages of "Ax_1C" and "Cx_1A" cannot be expressed as the language of a single sort regular pattern.

To illustrate this difficulty in comparisons more clearly, we found, through experiments, that a 4-mmg of sort regular pattern languages where 3 occurrences of sort symbols are allowed in each pattern to be somewhat comparable to an 8-mmg of regular pattern languages, in terms of accuracies in signal peptide prediction. These numbers do not appear to be intuitively derivable to us.

Nevertheless, we did find a measure that has a very good correlation with the accuracy values we obtained. This measure is the total number of strings in the language of a k-mmg of up to approximately the length of the longest sample used. We expected that such a count would give us a good indication of the amount of over-generalization that a k-mmg makes. (The computational complexity involved in such countings has been discussed in [11,9,13], and learning-related results of it can be found in [12]. We note also that such a counting can be done for many kinds of patterns studied in molecular biology [6,7], for instance the PROSITE patterns [4].)

What does the correlation between this measure and the accuracies of the k-mmgs tell us? We answer this at the end of the paper.

This paper is structured as follows. We first introduce the terminology required in our subsequent discussion. This is followed by a detailed description of our experimental setup, where we also make some remarks that have been left out in Takae *et. al.*'s earlier work. We then present results from the experiments and finally, discuss their significance.

2 Preliminaries

The symbol N denotes the set of natural numbers. For a set $\mathcal{A}$ we denote by $card(\mathcal{A})$ the cardinality of $\mathcal{A}$. A *word* over a non-empty alphabet $\mathcal{A}$ is a string of symbols taken from $\mathcal{A}$. The empty word is a null string and is denoted ε. For a set $\mathcal{A}$, the symbols $\mathcal{A}^*$, $\mathcal{A}^+$, and $\mathcal{A}^{\leq n}$ denote the sets of all the words, non-empty words, and words of length n or less over $\mathcal{A}$, respectively. For any sets $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} - \mathcal{B}$ denotes the set $\{x \mid x \in \mathcal{A} \text{ and } x \notin \mathcal{B}\}$.

Let Σ be a finite set of alphabets. A *sort symbol* is a letter associated with a non-empty set $S \subseteq \Sigma$, and is written $[S]$, or $[A_1 A_2 \ldots]$ where $A_1, A_2, \ldots$ is an enumeration of the elements of S. Π denotes the set $\{[S] | S \neq \emptyset \wedge S \subseteq \Sigma\}$. For any $A \in \Sigma$, $[A]$ is treated as equivalent to A, and hence Σ is treated as a proper subset of Π.

A *regular pattern* is a string over $\Sigma \cup \{*\}$, and a *sort regular pattern* is a string over $\Pi \cup \{*\}$, where the '$*$' symbol is called a *variable.* A *substitution* θ for a pattern p is a set of replacements for variables in p with patterns, and sort symbols with sort symbols. The image of a pattern p under a substitution θ is written $p\theta$. Every occurrence of a variable or sort symbol may be replaced independently. A sort symbol $[S]$ can only be replaced with another sort symbol $[S']$ where $S' \subseteq S$. We write $p \preceq q$ if $p = q\theta$ for some substitution θ. We write $p \prec q$ if $p \preceq q$ but not $q \preceq p$. Given two sets of patterns P and Q, we write $P \sqsubseteq Q$ if for each $p \in P$, $p \preceq q$ for some $q \in Q$, and $P \sqsubset Q$ just in case $P \sqsubseteq Q$ but not $Q \sqsubseteq P$. In the latter case, P is also said to be more *specific* than Q, and Q more *general* than P.

The *language* $L(p)$ *for a pattern* p is the set of all $w \in \Sigma^*$ where $w = p\theta$ for some substitution θ, that is, $L(p) = \{w \in \Sigma^* \mid w \preceq p\}$. For a set of patterns P, $L(P)$ is the union of $L(p)$ of each $p \in P$. Note that for any sets of patterns P, Q, if $P \sqsubseteq Q$ then $L(P) \subseteq L(Q)$ [15]. Given $k \in N$, a finite set $S \subseteq \Sigma^*$ and a class of patterns $\wp \sqsubseteq (\Pi \cup \{*\})^*$, a set of patterns $P \subseteq \wp$ is a *k-minimal multiple generalization* (or k-mmg) for S within $\wp$ if $card(P) \leq k$, $S \subseteq L(P)$, and there are no other set of up to k patterns Q in $\wp$ where $Q \sqsubset P$ and $S \subseteq L(Q)$. Note that for each k, $\wp$, and S there may be more than one such k-mmgs.

2.1 The MMG Algorithm

We introduce the **MMG** algorithm and our implementation of it in this section. Given a positive integer k, a set of strings S and a class of patterns $\wp$, the following is the listing for the **MMG** algorithm given in [3] for finding a k-mmg. All patterns in the following are implicitly elements of $\wp$.

```
MMG($k$, $S$)
  (1) Let $P \leftarrow \{``*"\}$.
  (2) Let $\Delta k \leftarrow k$.
  (3) While $\Delta k \geq 2$ and there exists $p \in P$ and
                $Q \sqsubseteq \{p\}$ of more than one pattern where
        (i) $S - L(P - \{p\}) \subseteq L(Q)$, but
        (ii) no $Q' \subset Q$ has $S - L(P - \{p\}) \subseteq L(Q')$ (i.e. no pattern in $Q$
             is redundant),
    (3.1) Replace each $q \in Q$ with a more specific $q' \prec q$ until any further
          such replacement will result in $S \not\subseteq L(P) - L(\{p\}) \cup L(Q)$.
    (3.2) Let $P \leftarrow P - \{p\} \cup Q$.
    (3.3) Let $\Delta k \leftarrow \Delta k - |Q| + 1$.
  (4) Output $P$.
```

A detailed description of the algorithm can be found in [3], here we only explain the parts which is needed in our discussion. Intuitively, the **MMG** algorithm starts with a most general pattern, and continues making the patterns more specific until no further refinement can be performed.

To make the search for Q at step (3) efficient, the **MMG** algorithm looks for candidates for Q from only a subset $\rho(p)$ of refinements of p of cardinality polynomial in the length of p. Such a ρ, called a *refinement operator*, is said to be *complete with respect to the class* $\wp$ just in case for all $p, p' \in \wp$, $p' \preceq p \Leftrightarrow p' \in \rho^+(p)$, where ρ^+ is the transitive closure of ρ. It has been shown that for a complete refinement operator ρ, for any p, such Q that fulfill the requirement at step (3) exists if and only if some $Q'' \subseteq \rho(p)$ also fulfills the requirement — hence limiting the search to $\rho(p)$ does not make the search less complete.

Refinement Operator. In this paper we use the pattern class where only up to a number of occurrences of the letters in $\Pi - \Sigma$ and $\{*\}$ are allowed in each pattern. (The regular patterns are simply patterns where 0 occurrence of the letters in $\Pi - \Sigma$ are allowed.) Let $\wp(m, s)$ denote the class of patterns with at most m occurrences of variables and at most s occurrences of the letters in $\Pi - \Sigma$ in each pattern. It is clear that for each m, $s \in N$, $\wp(m, s) \subset \wp(m+1, s)$ and $\wp(m, s) \subset \wp(m, s+1)$. For each class $\wp(m, s)$, we use the following as refinement operator.

```
$\rho(p)$
  (1) Let $P \leftarrow \emptyset$.
  (2) For each variable occurrence '$*$' in $p$ and
            for each $q$ in $\{$"$*[\Sigma]*$", "$*[\Sigma]$", "$[\Sigma]*$", $\varepsilon$ $\}$,
      (2.1) Let $p'$ be the pattern obtained from $p$ by replacing the '$*$' with $q$,
      (2.2) if $p' \in \wp(m, s)$, add $p'$ to $P$.
  (3) For each sort symbol occurrence $[S]$ in $p$, for each $S' \in \{S - \{X\} \mid X \in S\}$,
      (3.1) Add the pattern obtained by replacing $[S]$ with $[S']$ to $P$.
  (4) Output $P$.
```

Hence in this case, $card(\rho(p))$ is linear in the length of p, $|p|$ say, and furthermore, $\rho(p)$ can be computed in time $O(|p|)$.[4] We let $m = 4$ throughout this paper, that is, we consider only the classes $\wp(4, s)$ for some $s \in N$. For this reason we also write $\wp(4, s)$ simply $\wp(s)$.

Implementation. Our implementation of the algorithm is in C, and compiles on GCC 3.3 and 3.4 (on both `gcc` and `g++`). The program supports most of the modifications studied in the earlier publications [2,17,16], and provides other useful features such as the automatic compilation of alphabet from the input sample (that is, the alphabet for the amino acids is not coded into the program). The program can be downloaded from http://www.daisy.ai.kyutech.ac.jp/~kalngyk/.

3 Experimental Setup and Reproduced Earlier Results

Our samples are biosequences selected from the flat files for bacterial (release 137.0), plant (release 138.0) and rodent (release 138.0) sequences from NCBI GenBank [5]. All the signal peptide entries (that is, entries with the primary tag `sig_peptide`) were extracted, and from among these, we discarded the entries where the corresponding coding sequence (CDS)

1. has more than one translation table specified, or
2. translates into a sequence different from that given in the file.

We also removed the signal peptide entries which

1. translates into a sequence containing the '$*$' symbol, or
2. does not begin with a Methionine, or where
3. the location specified is fuzzy, or references an external sequence.

[4] Note that this operator is not complete with respect to any class $\wp(m, s)$ where $1 \le m \le 2$ and $s \ge 1$. As an example, let A, $C \in \Sigma$, $p =$"$*A[AC]A*$" and $q =$"$*$". Clearly, p, $q \in \wp(2, 1)$ and $p \preceq q$, but it can be shown that $p \notin \rho^+(q)$.

Finally, we remove the prefixing Methionine from each sequence obtained. We denote this set of sequences by $\mathtt{POS}_T$, where T is a sequence type of either bacterial, plant, or rodent. The following table lists the number of signal peptide sequences obtained using this method. Entries in brackets are numbers for the sequences used in [17] (corrected to count only distinct sequences) and the increase comes from the growth in the available sequences from GenBank.

Table 1. Sequences used in finding k-mmgs ($\mathtt{POS}_T$) and in evaluation ($\mathtt{POS}_T$, $\mathtt{NEG}_T$)

Sequence type T	$card(\mathtt{POS}_T)$	$\mathtt{POS}_T$ min/max/ave length	$card(\mathtt{NEG}_T)$
bacterial	1869 (422)	10 / 31 / 23	18690
plant	1694 (324)	10 / 30 / 22	16940
rodent	1803 (745)	10 / 30 / 21	18030

The negative examples $\mathtt{NEG}_T$ for each type T are (1) randomly chosen substrings of (randomly selected) type T CDSs, of a randomly decided length between 21 to 35, which are (2) not a substring of any sequence in $\mathtt{POS}_T$. While it is not guaranteed that sequences obtained this way will not turn out to be signal peptides, we trust that this odds is small, and by using a large number of sequences we can reduce its effects.

3.1 Measures for Evaluating a k-mmg

Accuracy. For a sequence type T, let $S \subset \mathtt{POS}_T$ be the samples used in deriving a k-mmg P. For each such P and S we define the following measures of accuracy,

1. The *positive accuracy*, $p(P,S) = 100\times\frac{card((\mathtt{POS}_T - S)\cap L(P))}{card(\mathtt{POS}_T - S)}$
2. The *negative accuracy*, $n(P) = 100\times\frac{card(\mathtt{NEG}_T - L(P))}{card(\mathtt{NEG}_T)}$
3. The *overall accuracy*, $acc(P,S) = \sqrt{p(P,S)\cdot n(P)}$

Coverage. For a k-mmg P and a length n, the quantity $card(L(P)^{\leq n})$ gives us an indication of the amount of over-generalization P has made within $\Sigma^{\leq n}$. In this paper we are interested in the case where $n = 30$, since that is roughly the longest length of our positive examples. Hence for a k-mmg P we define the *coverage of* P,

$$cov(P) = card(L(P)^{\leq 30})$$

In our experiment, to count $cov(P)$, we first construct a DFA [10] that accepts the words in $L(P)$, and then count the number of distinct paths of up to length 30 which reaches an accepting state [13].

3.2 Reproduced Earlier Results Using Updated Samples

We first reproduce the results as obtained in [17]. For each T and for each n in $\{50, 100, 150, \ldots, 800\}$, we randomly choose 100 sets of samples $S_1^n, S_2^n, \ldots, S_{100}^n$

each of size n from $\texttt{POS}_T$. From each such S_i^n we obtain a 5-mmg P_i^n from the class $\wp(9)$. We then compute the average accuracy $\sum_i acc(P_i^n, S_i^n)/100$, of the 100 5-mmgs obtained from samples of the size n, and we plot these accuracy values against n to see how this accuracy depends on the input size. These dependencies are quite similar to those observed in [17]. In our plots (Fig. 1) we also show the averages of $n(P_i^n)$ and $p(P_i^n, S_i^n)$ for a better picture of what leads to the trend observed in $acc(P_i^n, S_i^n)$. We see that the positive accuracy increases but negative accuracy decreases as we increase the sample size. This tells us that the k-mmgs produced becomes more general as we add more samples to **MMG**.

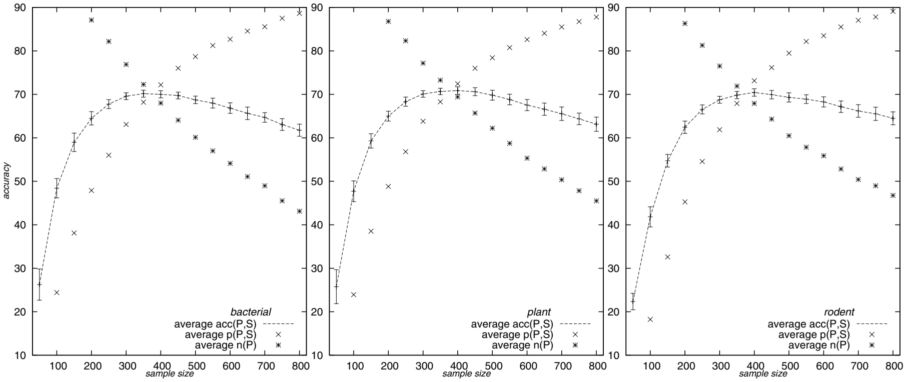


Fig. 1. Accuracies ($acc(P, S)$) vs input sample sizes ($card(S)$) of k-mmgs

4 Results: Correlation of Accuracies with Coverages

In this section we study the correlation between the accuracies of k-mmgs with their coverages, to examine the effects of over-generalization on the k-mmgs, under various conditions. We only reproduce the plots for bacterial samples here, but all the results in this section are also observed on the plant and rodent samples.

4.1 Coverages of k-mmgs from the Same k, $\wp(s)$ and Sample S

We first compare among the k-mmgs obtainable from the same sample. We run a modified form of the **MMG** algorithm to make it produce more than one k-mmg on a given sample set. This is done in a straight-forward manner, by letting **MMG** continue on a few different branches at step (3.1), each branch with a different order on the patterns in Q to refine.

Fig. 2 shows accuracies ($acc(P, S)$) against coverage ($cov(P)$) for the 5-mmgs obtained from respectively $\wp(1)$, $\wp(5)$ and $\wp(9)$, all from the same sample (of size 800). We see a very close correlation for the case of $\wp(1)$ and the correlation becomes less obvious for the case of $\wp(9)$, since the accuracies then conglomerate around similar values.

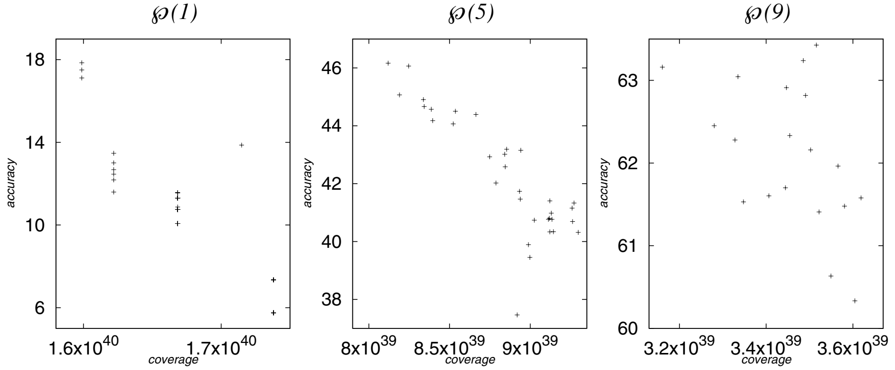


Fig. 2. $acc(P, S)$ vs $cov(P)$ of 5-mmgs obtained from the same sample

4.2 Coverages of k-mmgs for Various k, $\wp(s)$ at Different Sample Sizes

The correlation of accuracies with coverages can be observed across k-mmgs obtained from different samples as well as classes. Fig. 3 shows these values for k-mmgs obtained from the classes

1. 3-mmgs from $\wp(0)$ (legend ⊡)
2. 3-mmgs from $\wp(3)$ (legend ×)
3. 5-mmgs from $\wp(0)$ (legend △)
4. 5-mmgs from $\wp(5)$ (legend ○)
5. 5-mmgs from $\wp(9)$ (legend +)

In each graph in Fig. 3 we fix a sample size n, and for each of the classes we obtain 50 k-mmgs, each from a different sample of size n. Comparing among the graphs, we see how increasing the sample size makes the k-mmgs more general, and we also see how this affects their accuracies. The values obtained are in consistency with those in Section 4.1. Interestingly, k-mmgs of similar coverages almost always score about the same accuracies, regardless of k, or the class where the patterns are drawn. We also notice, from the plots for samples of the sizes 50∼150, that k-mmgs with coverages below a certain threshold score lower accuracies as their coverages decrease. We look at this region of the plots more closely in Section 4.4.

4.3 Similarly Accurate k-mmgs of Different k, $\wp(s)$ at Different Sample Sizes

Allowing letters in $\Pi - \Sigma$ to occur in patterns achieves similar effects as increasing the number k of patterns allowed in a k-mmg. However, given any k-mmg from $\wp(n)$ for some k and n, it is difficult to decide if some k'-mmgs from $\wp(n')$ for some $k' \neq k$ and $n' \neq n$ will be similarly accurate for protein prediction. Nevertheless, we found a specific case where such accuracies are comparable for all the sample sizes used in our tests: the 4-mmgs obtained from $\wp(3)$ (legend ⊡) and 8-mmgs from $\wp(0)$ (legend ×). This is shown in Fig. 4.

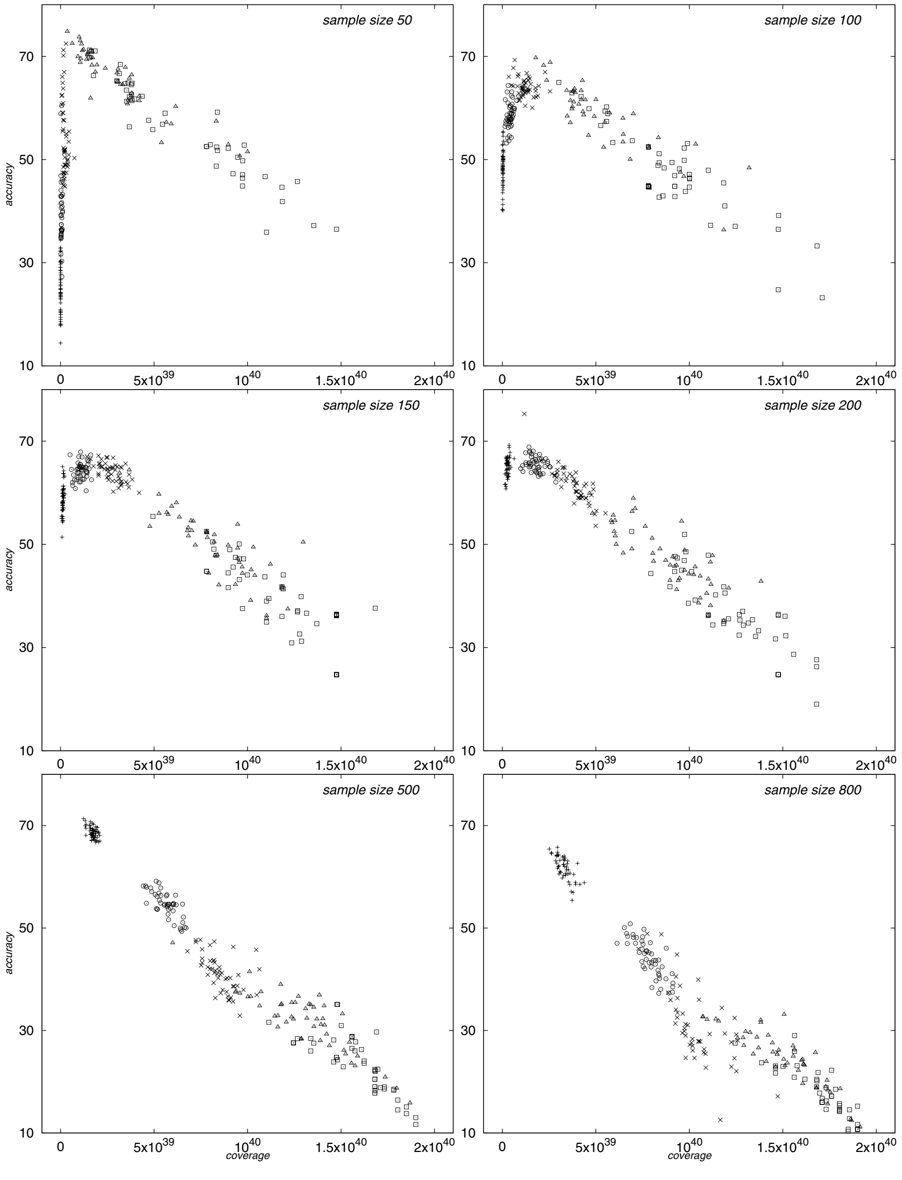


Fig. 3. $acc(P, S)$ vs $cov(P)$ of k-mmgs obtained from different classes and samples. $\boxdot$ = 3-mmgs from $\wp(0)$ $\times$ = 3-mmgs from $\wp(3)$ $\triangle$ = 5-mmgs from $\wp(0)$ $\bigcirc$= 5-mmgs from $\wp(5)$ $+$ = 5-mmgs from $\wp(9)$.

Interestingly, while the two different classes produce k-mmgs of similar accuracies, it is very much faster for the **MMG** algorithm to find 4-mmgs in $\wp(3)$ than 8-mmgs in $\wp(0)$, since the runtime of the algorithm increases more in k than in the size of the refinement operator.

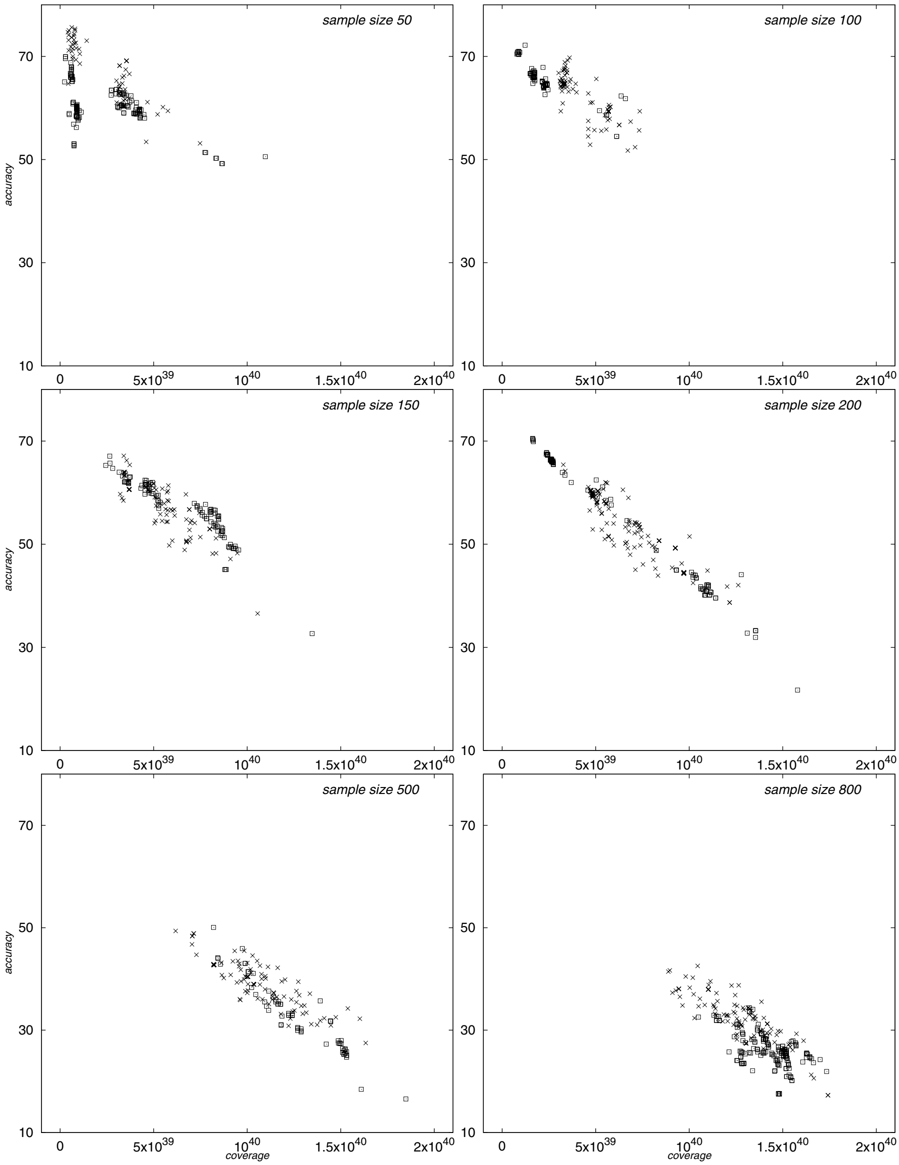


Fig. 4. $acc(P, S)$ vs $cov(P)$ of 4-mmgs from $\wp(3)$ and 8-mmgs from $\wp(0)$.
$\boxdot$ = 4-mmgs from $\wp(3)$ $\times$ = 8-mmgs from $\wp(0)$.

4.4 Overly Specific *k*-mmgs

We now show the details on the positive correlation between accuracies and coverage in the low coverage region in Fig. 3, by showing the x-axis in log scale.

In Fig. 5 we show respectively, the accuracies, the positive accuracies, and the negative accuracies of

(1) 5-mmgs from $\wp(4)$ (legend ⊡) (3) 5-mmgs from $\wp(9)$ (legend ○)
(2) 5-mmgs from $\wp(5)$ (legend ×) (4) 7-mmgs from $\wp(20)$ (legend ■)

obtained from samples of size 50.

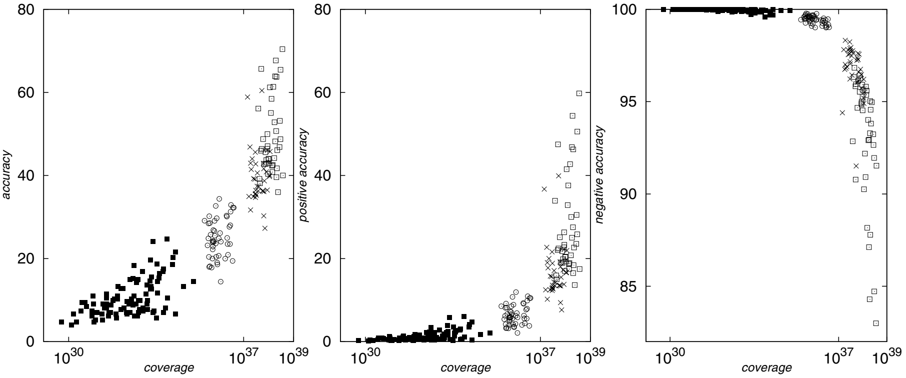


Fig. 5. Details of $acc(P, S)$, $p(P, S)$ and $n(P)$, against $cov(P)$ for samples of size 50

We see a rapid decline of the positive accuracies around the coverage range of 10^{30}~10^{38}, and for this region the trend of the accuracies follows closely that of the positive accuracies.

5 Conclusions

We now revisit the proposition made by Takae *et. al.* in [17], where they observed that the k-mmgs of sort regular patterns achieve higher accuracies than k-mmgs of regular patterns. We have shown that these higher accuracies are brought forth by a reduction of over-generalization in the k-mmgs of the sort regular patterns. We also argued that similar reduction in over-generalization can be achieved by increasing the number k of patterns allowed in a k-mmg, though we lack a theoretical basis to make any meaningful comparison between the two. Nevertheless, in Section 4.3 we gave an empirical case where k-mmgs of a higher k of only regular patterns scored similar accuracies as the k-mmgs of sort regular patterns. It is noteworthy, however, that the **MMG** algorithm finds k-mmgs of sort regular patterns very much faster than it finds k-mmgs of regular patterns of similar accuracies. This is because the runtime of the algorithm grows faster on k than the usage of letters in $\Pi - \Sigma$ (which results in a larger refinement operator).

By counting the coverages of the k-mmgs, that is, the number of strings of up to a certain length in the languages of the k-mmgs, we also found that k-mmgs with similar coverages always achieve similar accuracies — and this is regardless of the number k, or the kind of patterns used in the k-mmg. For the samples we used, we found that in general the accuracy values correlate negatively with the coverages, except for very small values of coverages, where the opposite trend is observed. This implies that in general, the k-mmgs obtained in our tests are over-generalized, and hence as a rule it is preferable to find more specific k-mmgs, although we should also be on the guard so that they do not become overly specific, for which we have seen that the accuracies will go into a rapid decline very suddenly. In practice, with respect to any samples of a specific kind, it may be possible for us to locate the range of coverage values where the k-mmgs' accuracies would be highest, and then use these coverage values as a guide in whether or not to accept an **MMG** output.

We note that such coverage may also be useful for making comparisons between pattern sets that are otherwise incomparable through syntax or set inclusion, a possibility which may be interesting in both theory and practice.

Finally, we hope that the idea of coverage can be adopted in the analysis of other types of patterns used to represent biological sequences, such as the PROSITE patterns [4].

Acknowledgement

We would like to thank the reviewers for helpful comments. Yen Kaow Ng is supported by the Japanese Government Scholarship of the Ministry of Education, Science, Sports, Culture and Technology of Japan. Hirotaka Ono is supported by the Scientific Grant in Aid of the Ministry of Education, Science, Sports, Culture and Technology of Japan.

References

1. D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.
2. H. Arimura, R. Fujino, T. Shinohara, and S. Arikawa. Protein motif discovery from positive examples by Minimal Multiple Generalization over regular patterns. In *Proceedings of the Genome Informatics Workshop*, pages 39–48, 1994.
3. H. Arimura, T. Shinohara, and S. Otsuki. Finding minimal generalizations for unions of pattern languages and its application to inductive inference from positive data. In *Proc. of the 11th Ann. Symp. on Theoretical Aspects of Comp. Sci. (STACS'94)*, volume 775 of *Lecture Notes in Computer Science*, pages 649–660. Springer-Verlag, 1994.
4. A. Bairoch. PROSITE: A dictionary of sites and patterns in proteins. *Nucl. Acids Res.*, 25(19):2241–2245, 1991.
5. D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. Genbank: update. *Nucl. Acids Res.*, 32(Database-Issue):23–26, 2004.
6. A. Brāzma, I. Jonassen, I. Eidhammer, and D. Gilbert. Approaches to the automatic discovery of patterns in biosequences. *J. Comp. Biol.*, 5(2):277–304, 1998.
7. B. Brejova, T. Vinar, and M. Li. *Pattern Discovery: Methods and Software*, chapter 29, pages 491–522. Humana Press, 2003.

8. J. Case, S. Jain, R. Reischuk, F. Stephan, and T. Zeugmann. Learning a subclass of regular patterns in polynomial time. In R. Gavaldà, K. P. Jantke, and E. Takimoto, editors, *Algorithmic Learning Theory: Fourteenth International Conference (ALT' 03)*, volume 2842 of *Lecture Notes in Artificial Intelligence*, pages 234–246, 2003.
9. C. Chan, M. Garofalakis, and R. Rastogi. RE-tree: an efficient index structure for regular expressions. *The VLDB Journal*, 12(2):102–119, 2003.
10. J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.
11. S. Kannan, Z. Sweedyk, and S. Mahaney. Counting and random generation of strings in regular languages. In *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 551–557. Society for Industrial and Applied Mathematics, 1995.
12. Y. K. Ng and T. Shinohara. Inferring unions of the pattern languages by the most fitting covers. In *Algorithmic Learning Theory: Sixteenth International Conference (ALT' 05).* to appear.
13. H. Ono and Y. K. Ng. Best fitting fixed-length substring patterns for a set of strings. In *Proceedings of The Eleventh International Computing and Combinatorics Conference (COCOON'05).* to appear.
14. A. Shinohara. String pattern discovery. In Shai Ben-David, John Case, and Akira Maruoka, editors, *Algorithmic Learning Theory: Fifteenth International Conference (ALT' 04)*, volume 3244 of *Lecture Notes in Artificial Intelligence*, pages 1–13, 2004.
15. T. Shinohara. Polynomial time inference of extended regular pattern languages. In *RIMS Symposia on Software Science and Engineering, Kyoto, Japan*, volume 147 of *Lecture Notes in Computer Science*, pages 115–127. Springer-Verlag, 1982.
16. T. Shinohara and Y. K. Ng. Strong biases for the minimal multiple generalization algorithm on samples of very small sizes. In *The Proceedings of the 57th Meeting of SIG-FPAI (November 2004).* The Japanese Society of Artificial Intelligence, 2004.
17. T. Takae, T. Kasai, H. Arimura, and T. Shinohara. Knowledge discovery in biosequences using sort regular patterns. Workshop on Applied Learning Theory, 1998.
18. J. Uemura and M. Sato. Compactness and learning of classes of unions of erasing regular pattern languages. In *Algorithmic Learning Theory: Thirteenth International Conference (ALT' 02)*, volume 2533, pages 293–307. Springer-Verlag, 2002.
19. M. Yamaguchi, S. Shimozono, and T. Shinohara. Finding minimal multiple generalization over regular patterns with alphabet indexing. In *Proceedings of the Seventh Workshop on Genome Informatics*, volume 7, pages 51–60. Universal Academy Press, 1996.

The q-Gram Distance for Ordered Unlabeled Trees*

Nobuhito Ohkura[1], Kouichi Hirata[2], Tetsuji Kuboyama[3], and Masateru Harao[2]

[1] Graduate School of Computer Science and Systems Engineering,
Kyushu Institute of Technology, Japan
ookura@dumbo.ai.kyutech.ac.jp

[2] Department of Artificial Intelligence, Kyushu Institute of Technology, Japan
{hirata, harao}@ai.kyutech.ac.jp

[3] Center for Collaborative Research, University of Tokyo, Japan
kuboyama@ccr.u-tokyo.ac.jp

Abstract. In this paper, we investigate *the q-gram distance for ordered unlabeled trees* (*trees*, for short). First, we formulate a *q-gram* as simply a tree with q nodes isomorphic to a line graph, and the *q-gram distance* between two trees as similar as one between two strings. Then, by using the *depth sequence* based on *postorder*, we design the algorithm *EnumGram* to enumerate all q-grams in a tree T with n nodes which runs in $O(n^2)$ time and in $O(q)$ space. Furthermore, we improve it to the algorithm *LinearEnumGram* which runs in $O(qn)$ time and in $O(qd)$ space, where d is the depth of T. Hence, we can evaluate the q-gram distance $D_q(T_1, T_2)$ between T_1 and T_2 in $O(q \max\{n_1, n_2\})$ time and in $O(q \max\{d_1, d_2\})$ space, where n_i and d_i are the number of nodes in T_i and the depth of T_i, respectively. Finally, we show the relationship between the q-gram distance $D_q(T_1, T_2)$ and the *edit distance* $E(T_1, T_2)$ that $D_q(T_1, T_2) \leq (gl + 1)E(T_1, T_2)$, where $g = \max\{g_1, g_2\}$, $l = \max\{l_1, l_2\}$, g_i is the degree of T_i and l_i is the number of leaves in T_i. In particular, for the *top-down tree edit distance* $F(T_1, T_2)$, this result implies that $D_q(T_1, T_2) \leq \min\{g^{q-2}, l - 1\}F(T_1, T_2)$.

1 Introduction

The *q-gram*, which is a string with length just q, is one of the bases for string processing, in particular, approximate string matching [8,16]. Then, the *q-gram profile* is formulated as a vector of the frequency for every q-gram, and the *q-gram distance* between two strings is as the difference of their q-gram profiles.

Such a q-gram (or also called n-gram) has also been widely applied to text indexing and filtering,data mining, web mining, and so on [4,6,7,12]. Note that, while their researches deal with semi-structured data, they always focus on the labels rather than the structures of the data and apply the q-gram for strings.

* This work is partially supported by Grand-in-Aid for Scientific Research 15700137, 16016275 and 17700138 from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 189–202, 2005.

On the other hand, it is necessary for filtering structured data and finding the frequency paths from structured data (*cf.*, [5,6,15]) to extract the structured features. Hence, in this paper, in order to characterize just the structures, we investigate the q-gram for *ordered unlabeled trees* which we call *trees* simply.

In this paper, first we adopt the simplest definition that a q-gram is a tree with q nodes isomorphic to a line graph. Then, we formulate the *q-gram profile* and the *q-gram distance* for trees as similar as ones for strings [16].

Next, by using the *depth sequence* [10,11] based on *postorder*, we design the naive and simple algorithm *EnumGram* to enumerate all q-grams in a tree T. After selecting the current depth as the depth of the left leaf of some q-gram, the algorithm *EnumGram* searches for the depth of the right leaf of the q-gram from the right side of the current depth in the depth sequence of T. For a tree T with n nodes, the algorithm *EnumGram* runs in $O(n^2)$ time and in $O(q)$ space.

Furthermore, we improve it to the algorithm *LinearEnumGram*, which checks whether or not the current depth is the right leaf or the root of some q-gram, by calling the two tables of size $O(qd)$, where d is the depth of T. Then, the algorithm *LinearEnumGram* runs in $O(qn)$ time and in $O(qd)$ space. This implies that, if we can regard q as a constant or as $q \ll n$, then the algorithm *LinearEnumGram* runs in linear time on n. Hence, we can evaluate the q-gram distance $D_q(T_1, T_2)$ between T_1 and T_2 in $O(q \max\{n_1, n_2\})$ time and in $O(q \max\{d_1, d_2\})$ space, where n_i and d_i are the number of nodes in T_i and the depth of T_i, respectively.

Note that, while the original depth sequence in [1,2,10,11,18] has been defined by using *preorder*, our depth sequence is defined by using *postorder*. This is essential for our algorithms to search for the depth of the leaf of a q-gram.

Finally, we investigate the relationship between the q-gram distance $D_q(T_1, T_2)$ and the *edit distance* $E(T_1, T_2)$ (*cf.*, [3,9,14,19]) between two trees T_1 and T_2. Here, the *edit distance* $E(T_1, T_2)$ is the minimum number of applications of the edit operation, that is, *insertion* and *deletion* (since we deal with just unlabeled trees) transforming from T_1 to T_2.

For two strings x and y, it is known that $D_q(x, y) \leq 2qE(x, y)$ [16]. In contrast, we show that $D_q(T_1, T_2) \leq (gl + 1)E(T_1, T_2)$, where $g = \max\{g_1, g_2\}$, $l = \max\{l_1, l_2\}$, g_i is the degree of T_i and l_i is the number of leaves in T_i. Furthermore, the *top-down tree edit distance* $F(T_1, T_2)$ [13,17] is the restricted edit distance that the edit operator can be applied to just leaves. Then, the above result implies that $D_q(T_1, T_2) \leq \min\{g^{q-2}, l - 1\}F(T_1, T_2)$.

2 The q-Gram Distance for Ordered Unlabeled Trees

A *tree* is a connected graph without cycles. A *rooted tree* is a tree with one node r chosen as its *root*. For a tree $T = (N, E)$, we sometimes denote $v \in T$ instead of $v \in V$, and $|T|$ instead of $|N|$.

For each node v in a rooted tree with the root r, let $UP_r(v)$ be the unique path from v to r. If $UP_r(v)$ has exactly d edges, then we say that the *depth* of

v is d and denote it by $dep(v) = d$. In particular, $UP_r(r) = \{r\}$ and $dep(r) = 0$. For a tree T, we denote $\max\{dep(v) \mid v \in T\}$ by $dep(T)$.

The *parent* of $v(\neq r)$ is its adjacent node on $UP_r(v)$ and the *ancestors* of $v(\neq r)$ are the nodes on $UP_r(v) - \{v\}$. The parent and the ancestors of the root r are undefined. We say that u is a *child* of v if v is the parent of u, and u is a *descendant* of v if v is an ancestor of u. A *leaf* is a node having no children. A node that is neither the root or the leaves is called an *internal node*. The number of all children of a node v is called a *degree* of v and denoted by $deg(v)$. For a tree T, we denote $\max\{deg(v) \mid v \in T\}$ by $deg(T)$ and the number of all leaves of T by $lvs(T)$.

A tree is *ordered* if a left-to-right order for the children of each node is given; *unordered* otherwise. Furthermore, we deal with a rooted ordered *unlabeled* tree, so we call it a tree simply.

Let T be a tree with the root v and the children $v_1, \dots, v_m$ of v. The *postorder traversal* (*postorder*, for short) of T is obtained by visiting v_i $(1 \leq i \leq m)$ in order, recursively, and then visiting v. For a tree T with n nodes, suppose that $v_1 \cdots v_n$ is the sequence of nodes of T in *postorder*. Then, the sequence $D(T) = dep(v_1) \cdots dep(v_n)$ is called the *depth sequence* of T [10,11]. For the depth sequence $D(T) = D$ of T, we denote $\max\{d \mid d \in D\}$ by $\max D$. Then, it is obvious that $dep(T) = \max D$.

Note that, while the original depth sequence in [1,2,10,11,18] has been defined by using *preorder*, our depth sequence is defined by using *postorder*. The reason is that, while the main topic in their works [1,2,10,11,18] is to enumerate *supertrees*, one in our work is to enumerate *subtrees*.

In this paper, we adopt the simplest definition of the q-gram for trees. We say that a *q-gram* is a tree with q nodes isomorphic to a *line graph* as an unrooted unordered tree, that is, the degree of the root is at most 2 and the degree of internal nodes is 1. It is obvious that the number of all q-grams is just $q - 1$. We denote the q-gram such that the first depth in its depth sequence (that is, the depth of the left leaf) is k by P_k $(1 \leq k \leq q-1)$. For example, Figure 1 describes all of the 4-grams P_1, P_2 and P_3 and their depth sequences.

Definition 1 (*cf.* Zhang & Shasha [19]). Let T and P be trees. Then, we say that P *matches* T *at a node* v if there exists a one-to-one mapping f from the nodes of P into the nodes of T satisfying the following conditions.

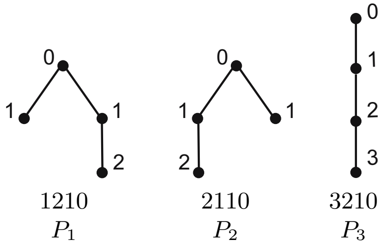


Fig. 1. All 4-grams and their depth sequences

1. f maps the root of P to v.
2. Suppose that f maps x to y and x has children $x_1, \ldots, x_l$ from left to right. Then, y has children $y_1, \ldots, y_m$ such that $m \geq l$ and there exists a monotone function $g : \{1, \ldots, l\} \rightarrow \{1, \ldots, m\}$ such that $f(x_i) = y_{g(i)}$ and $g(i_1) < g(i_2)$ whenever $i_1 < i_2$.

Then, we introduce the *q-gram profile* and the *q-gram distance* for trees based on the depth sequence, as similar as one for strings [16].

Definition 2. Let T be a tree and P_k a q-gram $(1 \leq k \leq q-1)$. Then, we say that P_k *occurs in* T if there exists a node v in T such that P_k matches T at v.

Let $G(T)[P_k]$ denote the total number of the occurrences of P_k in T. Then, the *q-gram profile* of T is a vector $G_q(T) = (G(T)[P_1], \ldots, G(T)[P_{q-1}])$.

Example 1. Consider the 4-gram profile of the trees T and S described as Figure 2. Note that the labels of T and S in Figure 2 denote the postorder traversals.

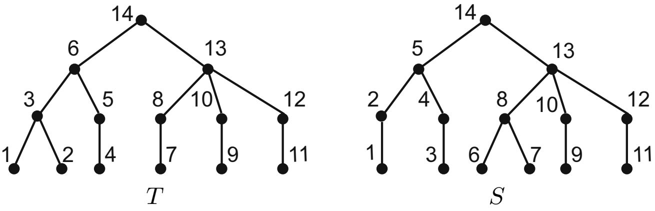


Fig. 2. The trees T and S in Example 1

Then, the following subtrees in T are corresponding to $G(T)[P_i]$.

P_1	$(3, 6, 5, 4), (6, 14, 13, 8), (6, 14, 13, 10), (6, 14, 13, 12),$ $(8, 13, 10, 9), (8, 13, 12, 11), (10, 12, 13, 11)$
P_2	$(1, 3, 6, 5), (2, 3, 6, 5), (3, 6, 14, 13), (5, 6, 14, 13),$ $(7, 8, 13, 10), (7, 8, 13, 12), (9, 10, 13, 12)$
P_3	$(1, 3, 6, 14), (2, 3, 6, 14), (4, 5, 6, 14), (7, 8, 13, 14),$ $(9, 10, 13, 14), (11, 12, 13, 14)$

Also the following subtrees in S are corresponding to $G(S)[P_i]$.

P_1	$(2, 5, 4, 3), (5, 14, 13, 8), (5, 14, 13, 10), (5, 14, 13, 12),$ $(8, 13, 10, 9), (8, 13, 12, 11), (10, 13, 12, 11)$
P_2	$(1, 2, 5, 4), (2, 5, 14, 13), (4, 5, 14, 13), (6, 8, 13, 10),$ $(6, 8, 13, 12), (7, 8, 13, 10), (7, 8, 13, 12), (9, 10, 13, 12)$
P_3	$(1, 2, 5, 14), (3, 4, 5, 14), (6, 8, 13, 14), (7, 8, 13, 14),$ $(9, 10, 13, 14), (11, 12, 13, 14)$

Hence, we can obtain the 4-gram profiles of T and S as $G_4(T) = (7, 7, 6)$ and $G_4(S) = (7, 8, 6)$, respectively.

Definition 3. Let T and S be trees and $q > 0$ be a positive integer. Then, the *q-gram distance* $D_q(T,S)$ between T and S is defined as follows.

$$D_q(T,S) = \sum_{k=1}^{q-1} |G(T)[P_k] - G(S)[P_k]| \,.$$

For the trees T and S in Example 1, it holds that $D_4(T,S) = 1$.

3 Enumeration Algorithm of All q-Grams in a Tree

In this section, we design the enumeration algorithm *EnumGram* of all q-grams in a tree T from the depth sequence $D = D(T)$ of T as Figure 3. Here, $D[i]$ is the i-th element of D and $P[k]$ is the number of the occurrences of P_k in T.

```
procedure EnumGram$(D,q)$
/* $D$: the depth sequence of a tree, $q > 0$: integer */
for $k = 1$ to $q-1$ do $P[k] \leftarrow 0$; /* initialize */
for $i = 1$ to $|D| - q$ do begin
    $k \leftarrow 1$; $j \leftarrow i+1$;
    while $j \leq |D|$ do begin
        if $D[j] = q + D[i] - k - 2$ then $P[k] \leftarrow P[k] + 1$;
        if $D[j] = D[i] - 1$ then $k \leftarrow k+1$; $i \leftarrow j$;
        if $k = q$ then break;
        $j \leftarrow j+1$;
    end /* while */
end /* for */
for $k = 1$ to $q-1$ do return $P[k]$;
```

Fig. 3. Algorithm *EnumGram*

Suppose that $|D| = n$. Then, the algorithm *EnumGram* counts the number of the occurrences of P_k in T by searching for the depth sequence D from $D[1]$ to $D[n]$. For the i-th iteration, *EnumGram* initializes k to 1 and $D[i]$ to the depth of the left leaf v of P_k, checks whether or not $D[j]$ is the depth of the right leaf of P_k for $i+1 \leq j \leq n$, updates k to $k+1$ and i to j if $D[j]$ is the depth of the parent of the current node with depth $D[i]$, and repeats the same procedure for the new k and $D[i]$ until $j > |D|$ or $k = q$.

Theorem 1. *The algorithm EnumGram(D,q) is correct.*

Proof. Consider the depth sequence $D = D(T)$ of a tree T and suppose that $|D| = n$. For a fixed i, let $D[i]$ be the current depth in D and w the current node in T corresponding to $D[i]$. Then, the algorithm *EnumGram* searches for P_k with the left leaf v as follows.

Initially, let $w = v$, $k = 1$ and $j = i + 1$. If *EnumGram* finds the depth $q + D[i] - k - 2$ as $D[j]$ in $D[i + 1] \cdots D[n]$, then it updates $P[k]$ to $P[k] + 1$, because $q + D[i] - k - 2$ is the depth of the right leaf of P_k with the left leaf v as shown bellow. Also if *EnumGram* finds the depth $D[i] - 1$ as $D[j]$, then there exists no more P_k with the left leaf v in T, because $D[i] - 1$ is the depth of the parent of w and the depth sequence is based on postorder. Hence, *EnumGram* updates k to $k + 1$ and i to j, that is, it finishes searching for P_k and begins searching for P_{k+1}, where $D[i]$ is updated to $D[j]$. This procedure for i is repeated until $j > |D|$ or $k = q$.

We show that, for the updated $D[i]$ and k, $q + D[i] - k - 2$ is the depth of the right leaf of P_k with the left leaf v. Note that w is an ancestor of v, and k is the number of edges between v and the parent r of w, or equivalently, $k = dep(v) - dep(r) + 1$. Furthermore, since the updated depth is $D[i]$, the algorithm *EnumGram* has been already searched for all of the descendants of w, so it searches for the depth of the descendants of r except the descendants of w.

Let u be the right leaf of P_k with the left leaf v. Since $|UP_r(v)| = k + 1$, P_k contains q nodes and r is contained by both $UP_r(v)$ and $UP_r(u)$, it holds that $|UP_r(u)| = q - k$. Then, the number of edges from r to u is $q - k - 1$, which means that $dep(u) - dep(r) = q - k - 1$. Since $dep(r) = D[i] - 1$, it holds that $dep(u) = q + D[i] - k - 2$. See Figure 4.

Since the above procedure is executed for each i $(1 \leq i \leq |D| - q)$, the algorithm *EnumGram* can enumerate all q-grams in T such that $D = D(T)$. □

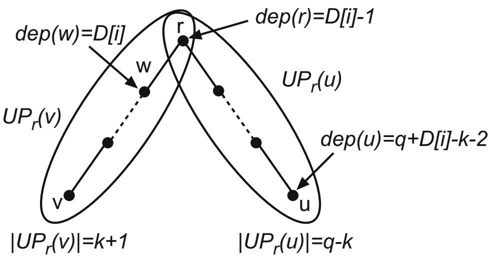


Fig. 4. The q-gram P_k in the proof of Theorem 1

Theorem 2. *The algorithm EnumGram(D, q) runs in $O(|D|^2)$ time and in $O(q)$ space.*

Note that we can obtain the q-gram distance between two trees T and S by using the results of *EnumGram*$(D(T), q)$ and *EnumGram*$(D(S), q)$. By Theorem 2, it holds the following corollary.

Corollary 1. *The q-gram distance $D_q(T, S)$ can be evaluated in $O(\max\{|T|^2, |S|^2\})$ time and in $O(q)$ space.*

4 Linear Time Enumeration Algorithm

Although the algorithm *EnumGram* can correctly enumerate all q-grams and is naive and simple, it is redundant, that is, it contains the similar running processes. Then, in this section, we improve the algorithm *EnumGram* to a faster algorithm *LinearEnumGram* as Figure 5.

```
procedure LinearEnumGram($(D, q)$)
/* $D$: the depth sequence of a tree, $q > 0$: integer */
/* initialize */
for $j = \max D$ downto 1 do
    for $k = 1$ to $\min\{q, j\}$ do
        $freq[j][k] \leftarrow 0$;
for $k = 1$ to $q - 1$ do $P[k] \leftarrow 0$;
for $d = \max D$ downto 0 do
    for $k = 1$ to $q - 1$ do
        if $0 \leq d - q + 1 + 2k \leq q$ then $count[d] \leftarrow count[d] \cup \{(d - q + 1 + 2k, k)\}$;
for $d = \max D - 1$ downto 1 do
    for $k = 1$ to $q - 1$ do
        if $0 \leq d + k \leq \max D$ then $shift[d] \leftarrow shift[d] \cup \{(d + k, k)\}$;
/* main routine */
for $i = 1$ to $|D|$ do begin
    foreach $(j, k) \in count[D[i]]$ do $P[k] \leftarrow P[k] + freq[j][k]$; /* Count */
    if $i = |D|$ then break;
    if $D[i] < \max D$ then
        foreach $(j, k) \in shift[D[i]]$ do
            $freq[j][k+1] \leftarrow freq[j][k+1] + freq[j][k]$; $freq[j][k] \leftarrow 0$; /* Shift */
    $freq[D[i]][1] \leftarrow freq[D[i]][1] + 1$;
end /* for */
for $k = 1$ to $q - 1$ do return $P[k]$;
```

Fig. 5. Algorithm *LinearEnumGram*

The main idea of the algorithm *LinearEnumGram* is to use two tables *count* and *shift*. For $0 \leq d \leq \max D$, $count[d]$ consists of the pairs (j, k) such that $j = d - q + 1 + 2k$, $0 \leq j \leq q$ and $1 \leq k \leq q - 1$. Also for $1 \leq d \leq \max D - 1$, $shift[d]$ consists of the pairs (j, k) such that $j = d + k$, $0 \leq j \leq \max D$ and $1 \leq k \leq q - 1$. Note that $|count[d]| \leq q$ and $|shift[d]| \leq q$. For example, the tables *count* and *shift* for $q = 4$ and $\max D = 3$ are described as Figure 6.

The following lemmas characterize the parameters j, k and d such that $(j, k) \in count[d]$ and $(j, k) \in shift[d]$, respectively.

Lemma 1. *Let P_k be a q-gram $(1 \leq k \leq q - 1)$ and d the depth of the right leaf (or the root if $k = q - 1$) of P_k. Then, the depth of the left leaf of P_k is $d - q + 1 + 2k$.*

d	(j, k)
3	$(2, 1)$
2	$(3, 2), (1, 1)$
1	$(2, 2)$
0	$(3, 3)$

count

d	(j, k)
2	$(3, 1)$
1	$(2, 1), (3, 2)$

shift

Fig. 6. The tables *count* and *shift* for $q = 4$ and $\max D = 3$

Proof. Let v, u and r be the the left leaf, the right leaf and the root of P_k, respectively. Then, it holds that $|\,UP_r(v)| = k + 1$ and $|\,UP_r(u)| = q - k$. Since $dep(u) = d$, it holds that $d - dep(r) = (q - k) - 1$, so $dep(r) = d + q + 1 + k$. Since $|\,UP_r(v)| = k+1$, it holds that $dep(v) = (d+q+1+k)+(k+1)-1 = d-q+1+2k$. See Figure 7 (left). □

Lemma 2. *Let P_k be a q-gram $(1 \leq k \leq q - 1)$ and d the depth of the root of P_k. Then, the depth of the root of P_k is $d + k$.*

Proof. Let v and r be the left leaf and the root of P_k, respectively. Since $|\,UP_r(v)| = k + 1$ and $dep(r) = d$, it holds that $dep(v) = d + (k + 1) - 1 = d + k$. See Figure 7 (right). □

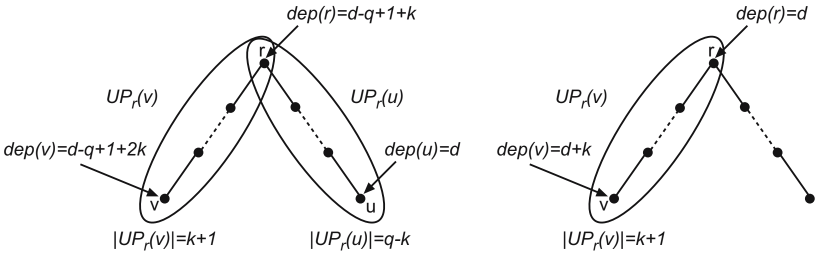


Fig. 7. The q-gram P_k in the proofs of Lemma 1 (left) and Lemma 2 (right)

Theorem 3. *The algorithm LinearEnumGram(D, q) is correct.*

Proof. Let $D = D(T)$ be the depth sequence of T and $D[i]$ the current depth in D. Note that *freq*$[j][k]$ is the number of the occurrences of P_k with the left leaf of the depth j in the depth sequence $D[1] \cdots D[i - 1]$. Then, by using *freq*$[j][k]$, the algorithm *LinearEnumGram* checks whether or not $D[i]$ is the depth of the right leaf of P_k, and, if so, then it updates the number of the occurrences of P_k by *freq*$[j][k]$.

Consider the "Count" routine. By Lemma 1 and the definition of *count*$[d]$, $(j, k) \in$ *count*$[d]$ implies that d and j are the depth of the left and right leaves of P_k, respectively. Hence, for every $(j, k) \in$ *count*$[D[i]]$, since $D[i]$ is the depth of the right leaf of P_k, *LinearEnumGram* updates $P[k]$ to $P[k] +$ *freq*$[j][k]$.

Consider the "Shift" routine. By Lemma 2 and the definition of $\mathit{shift}[d]$, $(j,k) \in \mathit{shift}[d]$ implies that d and j are the depth of the root and the left leaf of P_k, respectively. Then, for every $(j,k) \in \mathit{shift}[D[i]]$, there exists no more P_k with the left leaf v in T, because $D[i]$ is the depth of the root of P_k and the depth sequence is based on postorder. Hence, by shifting k to $k+1$, *LinearEnumGram* updates $\mathit{freq}[j][k+1]$ to $\mathit{freq}[j][k+1] + \mathit{freq}[j][k]$ and initializes $\mathit{freq}[j][k]$ to 0, that is, it finishes searching for P_k and begins searching for P_{k+1}.

After the above "Count" and "Shift" routines, *LinearEnumGram* updates $\mathit{freq}[D[i]][1]$ to $\mathit{freq}[D[i]][1] + 1$, in order to add $D[i]$ to the left leaf of P_1.

Since $\mathit{freq}[j][k]$ such that $(j,k) \in \mathit{count}(D[i])$ is the number of the occurrences of P_k of which depth of the left and right leaves are j and $D[i]$, and by summing up $\mathit{freq}[j][k]$ for every $D[i]$ in the "Count" routine, the algorithm *LinearEnumGram* can enumerate all q-grams in T such that $D = D(T)$. □

Theorem 4. *The algorithm LinearEnumGram*(D,q) *runs in* $O(q|D|)$ *time and in* $O(q \max D)$ *space.*

Proof. Note that, while the sizes of the tables *count* and *shift* are $O(q \max D)$, the "Count" and "Shift" routines in *LinearEnumGram* call just $\mathit{count}[D[i]]$ and $\mathit{shift}[D[i]]$, respectively, both of which sizes are at most $O(q)$ for every i. □

Hence, if we regard q as a constant or as $q \ll |D|$, which is a natural setting, then the algorithm *LinearEnumGram*(D,q) runs in $O(|D|)$ time and in $O(\max D)$ space. Furthermore, since $\max D = \mathit{dep}(T)$ for a tree T and its depth sequence $D = D(T)$, the following corollary also holds.

Corollary 2. *The* q*-gram distance* $D_q(T,S)$ *can be evaluated in* $O(q \max\{|T|,|S|\})$ *time and in* $O(q \max\{\mathit{dep}(T), \mathit{dep}(S)\})$ *space.*

Example 2. Consider the trees T and S in Example 1 again, where:

$$D(T) = 33232132323210,$$
$$D(S) = 32321332323210.$$

The tables *count* and *shift* for $q = 4$ and $\max D = 3$ have been already described as Figure 6. Then, Figure 8 describes the transitions of the table *freq* of the algorithm *LinearEnumGram* for $D(T)$ and $D(S)$, respectively, where 0 is omitted.

Note that the i-th column in Figure 8 denotes the table *freq* for the i-th iteration of the for-loop. Also the underlined number in the i-th column is added to $P[k]$ by the "Count" routine in the $(i+1)$-th iteration of the for-loop. Furthermore, the arrow $\searrow$ in the i-th column denotes the shifting by the "Shift" routine in the $(i+1)$-th iteration of the for-loop.

By summing the underlined numbers for every k ($k = 1,2,3$), we can obtain the q-gram profile of T and S as $G_4(T) = (7,7,6)$ and $G_4(S) = (7,8,6)$, respectively. Hence, it holds that $D_q(T,S) = 1$.

	j	k	3	3	2	3	2	1	3	2	3	2	3	2	1	0
	3	1	1	$2\searrow$		$1\searrow$			$1\searrow$		$1\searrow$		$1\searrow$			
	3	2			2	$\underline{2}$	$3\searrow$			1	$\underline{1}$	2	$\underline{2}$	$3\searrow$		
T	3	3						3	3	3	3	3	3	3	$\underline{6}$	6
	2	1			$\underline{1}$	1	$2\searrow$			$\underline{1}$	1	$\underline{2}$	2	$3\searrow$		
	2	2						2	2	2	2	2	2	$\underline{2}$	5	5
	1	1						1	$\underline{1}$	1	$\underline{1}$	1	$\underline{1}$	1	2	2

	j	k	3	2	3	2	1	3	3	2	3	2	3	2	1	0
	3	1	$1\searrow$		$1\searrow$			1	$2\searrow$		$1\searrow$		$1\searrow$			
	3	2		1	$\underline{1}$	$2\searrow$				2	$\underline{2}$	3	$\underline{3}$	$4\searrow$		
S	3	3					2	2	2	2	2	2	2	2	$\underline{6}$	6
	2	1		$\underline{1}$	1	$2\searrow$				$\underline{1}$	1	$\underline{2}$	2	$3\searrow$		
	2	2					2	2	2	2	2	2	2	$\underline{2}$	5	5
	1	1					1	1	$\underline{1}$	1	$\underline{1}$	1	$\underline{1}$	1	2	2

Fig. 8. The transitions of the table *freq* for T and S

5 The Relationship to Edit Distance

In this section, we discuss the relationship between the q-gram distance and the *edit distance* (*cf.*, [3,9,14,19]) for trees.

The *edit operation* on a(n unlabeled) tree T is one of the following two operations[1].

1. *Insertion* of a new node v as a child of a node $u \in T$.
2. *Deletion* of either an internal node or a leaf v from T, moving all children of v right under the parent of v.

In particular, we assume that every edit operation has a *unit cost* [14]. For trees T and S, the *edit distance* $E(T,S)$ between T and S is the minimum number of applications of the edit operation transforming from T to S.

As stated in Section 1, it is known the relationship $D_q(x,y) \leq 2qE(x,y)$ [16] between the q-gram distance $D_q(x,y)$ and the edit distance $E(x,y)$ for two strings x and y. However, this relationship does not hold for trees in general.

Example 3. Consider the trees T and S in Figure 9. Then, it is obvious that $E(T,S) = 1$. On the other hand, since $G_3(T) = (11,5)$ and $G_3(S) = (15,0)$, it holds that $D_3(T,S) = 9$. Then, it holds that $D_3(T,S) = 9 > 2 \cdot 3 \cdot 1 = 2qE(T,S)$.

Theorem 5. *Let* $g = \max\{deg(T), deg(S)\}$ *and* $l = \max\{lvs(T), lvs(S)\}$ *for trees* T *and* S. *Then, the following statement holds.*

$$D_q(T,S) \leq (gl+1)E(T,S).$$

[1] For labeled trees, the edit operation consists of insertion, deletion and *relabeling* [3,19].

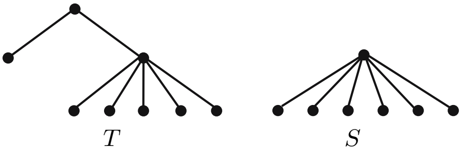


Fig. 9. Trees T and S in Example 3

Proof. Suppose that $q \geq 3$ and T is transformed to S by the deletion, that is, deleting an internal node $v \in T$. Let D_1, D_2 and D be the set of all q-grams containing v as a root, a leaf and an internal node, respectively. Consider whether or not a q-gram $P \in D_1 \cup D_2 \cup D$ is still in S.

1. Each $P \in D_1$ is also a q-gram in S.
2. For each $P \in D_2$, P occurs in S as $deg(v)$ q-grams.
3. Consider $P \in D(= D_3 \cup D_4 \cup D_5)$.
   - D_3: If P contains a leaf of T as a descendant of v, then P is *not* a q-gram in S.
   - D_4: Else if the parent of v is the root of P and the degree of it is 1 (that is, $P = P_{q-1}$), then P is *not* a q-gram in S, because such a P has been already counted as a q-gram in S by D_1.
   - D_5: Otherwise, P is also a q-gram in S.

Let $|D_i|$ be d_i and e the number of q-grams not containing v in T. Then, the number of all q-grams in T is $e + d_1 + d_2 + d_3 + d_4 + d_5$ and the number of all q-grams in S is $e + d_1 + deg(v)d_2 + d_5$. Hence, the following statement holds.

$$D_q(T, S) = (deg(v) - 1)d_2 + d_3 + d_4.$$

Let $g = deg(T)$ and $l = lvs(T)$. Note that d_2 is bounded by the number of all q-grams of which root is in $UP_u(v) - \{v\}$ for u such that $|UP_u(v)| = q$. The total number of all q-grams of which root is in $UP_u(v) - \{v\}$ is at most $l - 1$, so it holds that $d_2 \leq l - 1$. Furthermore, it is obvious that $d_3 \leq l$ and $d_4 \leq g$. Hence, the following statement holds.

$$D_q(T, S) \leq (g - 1)(l - 1) + g + l = gl + 1.$$

For the case deleting a leaf $v \in T$, let u be the parent of v. Then, the q-grams in T containing u and v are *not* q-grams in S, so $D_q(T, S)$ coincides with the number of such q-grams. By the bound of d_2, it holds that $D_q(T, S) \leq l - 1$. Furthermore, it is obvious that $D_2(T, S) = 1$.

Suppose that T is transformed to S by the insertion, that is, inserting v in S as an internal node. By replacing $v \in T$ with $v \in S$ and by replacing D_1, D_2 and $D(= D_3 \cup D_4 \cup D_5)$ with the set of all q-grams that v is a root, a leaf and an internal node, respectively, in S, we obtain the q-gram distance between T and S by letting $g = deg(S)$ and $l = lvs(S)$. Furthermore, the case inserting v in S as a leaf of S, let u be the parent of v. Since the q-grams in S containing u and v are *not* q-grams in T, it holds the similar result of deleting a leaf $v \in T$.

Let $g = \max\{deg(T), deg(S)\}$ and $l = \max\{lvs(T), lvs(S)\}$. Since the above estimation is corresponding to $E(T, S) = 1$, the statement holds. □

The *top-down tree edit distance* (or 1-*degree tree edit distance*) [13,17] is the restricted edit distance that the edit operator can be applied to just leaves. We denote the top-down tree edit distance between T and S by $F(T, S)$. Then, by using the proof of Theorem 5 when v is a leaf, the following corollary holds.

Corollary 3. *Let $g = \max\{deg(T), deg(S)\}$ and $l = \max\{lvs(T), lvs(S)\}$ for trees T and S. Then, the following statement holds for $q \geq 3$.*

$$D_q(T, S) \leq \min\{g^{q-2}, l-1\}F(T, S).$$

Proof. By Theorem 5, it holds that $D_q(T, S) \leq (l-1)F(T, S)$. Consider the case d_2 in the proof of Theorem 5 again. Suppose that $r \in UP_u(v) - \{v\}$ is the root of a q-gram P such that $dep(v) - dep(r) = k$ $(1 \leq k \leq q-1)$. Since $deg(r) \leq g$ and $|UP_r(v)| = k+1$, the number of all q-grams with a root r and a leaf v is at most $(g-1)g^{q-k-2}$. In particular, if $k = q-1$, then the number is 1. Then, d_2 is bounded by:

$$1 + \sum_{k=1}^{q-2}(g-1)g^{q-k-2} = 1 + (g-1)\sum_{k=0}^{q-3} g^k = g^{q-2}.$$

Hence, the statement holds. □

6 Conclusion

In this paper, we have investigated the q-gram for ordered unlabeled trees. First, we have formulated a q-gram as a tree with q nodes isomorphic to a line graph. Then, we have also formulated the q-gram profile and the q-gram distance for trees as similar as ones for strings [16].

Next, by using the depth sequence based on postorder, we have designed the naive and simple algorithm *EnumGram* to enumerate all q-grams in a tree T with n nodes that runs in $O(n^2)$ time and in $O(q)$ space. Furthermore, we have improved it to the algorithm *LinearEnumGram* that runs in $O(qn)$ time and in $O(qd)$ space. Here, d is the depth of T. Hence, we have evaluated the q-gram distance $D_q(T_1, T_2)$ between T_1 and T_2 in $O(q\max\{n_1, n_2\})$ time and in $O(q\max\{d_1, d_2\})$ space, where n_i and d_i are the number of nodes in T_i and the depth of T_i, respectively.

Finally, for the edit distance $E(T_1, T_2)$ and the top-down tree edit distance $F(T_1, T_2)$, we have shown that $D_q(T_1, T_2) \leq (gl+1)E(T_1, T_2)$ and $D_q(T_1, T_2) \leq \min\{g^{q-2}, l-1\}F(T_1, T_2)$, where $g = \max\{g_1, g_2\}$, $l = \max\{l_1, l_2\}$, g_i is the degree of T_i and l_i is the number of leaves in T_i.

The setting in this paper is closely related to extract structured features from structured data, such as filtering structured data and finding the frequency paths from structured data, for example, [5,6,15]. It is a future work to apply our algorithms to such an application.

The main idea in our algorithms is to maintain just the information of the depth of nodes. In order to extend our method to labeled trees, it is necessary to maintain the information of not only the depth but also the label of any node. Then, it is an important future work to design the algorithm to enumerate all labeled q-grams in a labeled tree efficiently. Furthermore, it is also a future work to investigate the relationship between the q-gram distance and the edit distance for ordered labeled trees.

The efficiency of our algorithms follows that we have adopted the simplest definition of a q-gram that is a tree with q nodes isomorphic to a line graph. On the other hand, it is also a natural definition that a q-gram is a tree with q nodes. However, this definition makes the enumeration of all q-gram more difficult, because of the existence of internal nodes with greater than degree 1. Hence, it is a future work to design the algorithm to enumerate all of such q-grams efficiently.

References

1. T. Asai, K. Abe, S. Kawazoe, H. Arimura, H. Sakamoto, S. Arikawa: *Efficient substructure discovery from large semi-structured data*, Proc. SDM'02, 2002.
2. T. Asai, H. Arimura, S. Nakano, T. Uno: *Discovering frequent substructures in large unordered trees*, Proc. DS'03, LNAI **2843**, 47–61, 2003.
3. P. Bille: *A survey on tree edit distance and related problems*, Theor. Comput. Sci. **337**, 217–239, 2005.
4. S. Burkhardt, J. Karkkainen: *Better filtering with gapped q-grams*, Proc. CPM'01, LNCS **2089**, 73–85, 2001.
5. K. Furukawa, T. Uchida, K. Yamada, T. Miyahara, T. Shoudai, Y. Nakamura: *Extracting characteristic structures among words in semistructured documents*, Proc. PAKDD'02, LNAI **2336**, 351–360, 2002.
6. M. Garofalakis, A. Kumar: *Correlating XML data streams using tree-edit distance embeddings*, Proc. PODS'03, 143–154, 2003.
7. D. Ikeda, Y. Yamada, S. Hirokawa: *Eliminating useless parts in semi-structured documents using alternation counts*, Proc. DS'01, LNAI **2226**, 113-127, 2001.
8. P. Jokinen, E. Ukkonen: *Two algorithms for approximate string matching in static texts*, Proc. MFCS'91, LNCS **520**, 240–248, 1991.
9. T. Kuboyama, K. Shin, T. Miyahara, H. Yasuda: *A theoretical analysis of alignment and edit problems for trees*, Proc. ICTCS'05, LNCS, 2005 (to appear).
10. S. Nakano, T. Uno: *Efficient generation of rooted trees*, National Institute of Informatics Technical Report NII-2003-005E, 2003.
11. S. Nakano, T. Uno: *Constant time generation of trees with specified diameter*, Proc. WG'04, LNCS **3353**, 33–45, 2004.
12. G Navarro, E Sutinen, J Tanninen, J Tarhio: *Indexing text with approximate q-grams*, Proc. CPM'00, LNCS **1848**, 350-363, 2000.
13. S. M. Selkow: *The tree-to-tree editing problem*, Inform. Proc. Let. **6**, 184–186, 1997.
14. D. Shasha, K. Zhang: *Fast algorithms for the unit cost edit distance between trees*, J. Algo. **11**, 581–621, 1990.
15. T. Uchida, T. Mogawa, Y. Nakamura: *Finding frequent structural features among words in tree-structured documents*, Proc. PAKDD'04, LNAI **3056**, 351–360, 2004.

16. E. Ukkonen: *Approximate string-matching with q-grams and maximal matches*, Theor. Comput. Sci. **92**, 191–211, 1993.
17. W. Yang: *Identifying syntactic differences between two programs*, Software–Practice and Experience **21**, 739–755, 1991.
18. M. J. Zaki: *Efficiently mining frequent trees in a forest*, Proc. SIGKDD'02, 71–80, 2002.
19. K. Zhang, D. Shasha: *Tree pattern matching*, in: A. Apostolico, Z. Galil (eds): *Pattern matching algorithms*, 341–371, 1997.

Monotone Classification by Function Decomposition

Viara Popova[1] and Jan C. Bioch[2]

[1] Department of Artificial Intelligence, Vrije Universiteit,
De Boelelaan 1081a, 1081HV Amsterdam, The Netherlands
popova@few.vu.nl

[2] Econometric Institute, Erasmus University Rotterdam,
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
bioch@few.eur.nl

Abstract. The paper focuses on the problem of classification by function decomposition within the frame of monotone classification. We propose a decomposition method for discrete functions which can be applied to monotone problems in order to generate a monotone classifier based on the extracted concept hierarchy. We formulate and prove a criterion for the existence of a positive extension of the scheme $f = g(S_0, h(S_1))$ in the context of discrete functions. We also propose a method for finding an assignment for the intermediate concept with a minimal number of values.

1 Introduction

Problem decomposition approaches are used in many areas of science, e.g. switching theory, game theory, reliability theory, machine learning. One of the applications in machine learning is in structured induction which aims at splitting a concept to be learnt in a hierarchy of sub-concepts which can be used separately to generate classification rules. The methods vary but the majority of them involve a human expert who provides domain knowledge of the structure of the problem. This process can take a long time and a lot of energy while the availability of the expert is not necessarily guaranteed.

The contribution of this paper is within the research in automating the decomposition process. We look at the problem from the point of view of classification for monotone data sets. A data set with a set of attributes A and a labeling λ is called *monotone* if the values of each attribute are ordered and for each two data points x, y such that $x \leq y$ (y dominates x on all attributes, i.e. $\forall i\ x_i \leq y_i$) it is true that $\lambda(x) \leq \lambda(y)$.

In practice monotone problems appear in various domain areas (e.g. credit rating, bankruptcy prediction, bond rating, etc.). For example in credit rating if one applicant for credit outperforms another on all criteria then he should be given at least the same chance for being approved. Here as well as in other areas it is desirable to use a monotone classifier not only with the aim to avoid

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 203–214, 2005.

unnecessary money loss but also to help in motivating the decision in front of internal or external parties.

The property of monotonicity appears in different settings in many areas of science - from natural language processing (upward/downward monotonicity of quantifiers) to game theory (co-operative games), reliability theory (semi-coherent structures), database theory (minimal keys), rough sets theory (reducts) and association rules (frequent patterns).

Monotone classification has been studied in the context of logical analysis of data, decision trees, decision lists, neural networks, rough sets theory, instance-based learning, etc. (see [6] for a list of references). In this paper we aim at building a decomposition hierarchy that preserves the monotonicity property with the ultimate goal of generating a monotone classifier.

The paper is organized as follows. Section 2 reviews related research. Section 3 presents our contribution. First experimental results are shown in Section 4. Section 5 concludes and gives some future research directions.

2 Related Research on Function Decomposition

Within machine learning the first attempts in decomposition were presented in [8] where manually defined concept structures were used with two layers of intermediate concepts. A related field to function decomposition is feature extraction which aims at constructing new better attributes from the existing ones. These methods in general need to be given in advance the operators they can use for generating the new attributes. Other approaches use an expert to decompose the problem and construct the concept hierarchy, e.g. [9] and [4]. The hierarchical structure is used to generate decision rules.

A lot of research has been done in the specific case of Boolean functions decomposition. Important results relevant for our approach were presented in [5] which investigates the problem of decomposability of partially defined Boolean functions. It concentrates on the complexity of determining whether a Boolean function is decomposable using a specific scheme. The most general scheme considered is:

$$f = g(S_0, h_1(S_1), \ldots, h_k(S_k))$$

where S_i are (not necessarily disjoint) subsets of the set of attributes A such that $\bigcup_{i=0}^{k} S_i = A$. The authors of [5] show that deciding whether a partially defined Boolean function is decomposable is an NP-complete problem for $k \geq 2$. For $k = 1$ the time complexity is $O(mn)$ where m is the number of data points and n is the number of attributes. The article also examines the problem of positive schemes in the frames of Boolean functions (see Section 3).

The research presented in this paper is related to the algorithm presented in [10,11]. This algorithm recursively decomposes a discrete function $y = f(A)$ into $y = g(S_0, h(S_1))$ (see Figure 1) where S_0 and S_1 are disjoint and $S_0 \cup S_1 = A$. The functions g and h are not predefined in the application of the method and are induced during the decomposition process. The requirement for them is to have

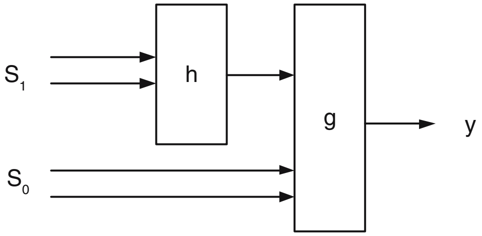


Fig. 1. The decomposition of a function

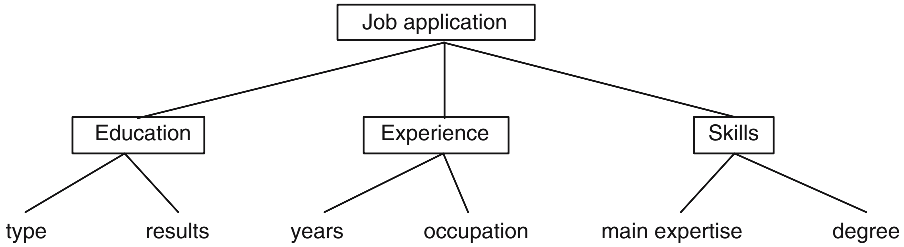


Fig. 2. A concept hierarchy

joint complexity which is lower than the complexity of f and that is determined using some complexity measure.

By applying the method recursively on h and g we can generate a hierarchy of concepts. Figure 2 shows an example of a concept hierarchy. The names of the intermediate concepts are not given by the decomposition method – they are just interpretations. The concepts in the leaves correspond to the attributes in A. They are grouped in more general concepts in the internal nodes which correspond to the intermediate functions generated by the algorithm.

The decomposition method proceeds as follows. *The attribute partition selection step* determines which is the best partition into S_0 and S_1 using a predefined complexity measure. *The basic decomposition step* finds the functions g and h for the partition. The procedure used is equivalent to graph coloring. *The overall function decomposition step* is the recursive application of the above two steps until no decomposition can be found such that the resulting functions are less complex than the function being decomposed.

Let us concentrate on the basic decomposition step of the algorithm. It starts by constructing the co-called *partition matrix*. The rows of this matrix correspond to the distinct values of the attributes in S_0 and similarly the columns correspond to the distinct values of the attributes in S_1. The entries of the matrix contain the class label for the specific combination of S_0 and S_1 values from the row and the column. Obviously some of these will be empty and considered as "don't care".

As an example of a partition matrix we consider the data set from Table 2. Let us choose the partition into the two subsets $S_0 = \{a_1, a_2, a_3, a_6\}$ and

$S_1 = \{a_4, a_5\}$. The corresponding partition matrix is given in Table 1 where the rows are labelled with the values of S_0 and the columns are labelled with the values of S_1.

Table 1. An example of a partition matrix

	13	23	11	12	22
3221	3	*	*	*	*
2221	*	3	*	*	*
3132	3	*	*	*	*
2112	*	*	2	*	*
2231	*	*	*	2	*
1121	*	*	1	*	2
1211	*	*	1	*	*
h	2	2	2	2	1

Two columns of the matrix are called *compatible* if they do not contradict each other or, more precisely, if they do not contain entries for the same row that are non-empty and are labelled with a different class label. The number of such pairs of entries is called the *degree of compatibility* of the two columns. If two columns are not compatible, they are called *incompatible*. In our example we have two incompatible columns: 11 and 22. For them the values of the attributes in S_0 are the same but the class label is different.

In order to find a new intermediate concept for S_1 we need a labelling for the columns of the partition table (in other words, values for the intermediate concept) such that g and h are consistent with f. This is exactly the case when incompatible columns are never assigned the same label. The resulting problem is equivalent to graph colouring and the corresponding graph is the so called *incompatibility graph*. Its vertices correspond to the columns of the partition matrix and there is an edge between two vertices if and only if their columns are incompatible.

In the example, the graph contains five vertices and only one edge between 11 and 22 which indicates that they should be assigned different colours. For these two columns there is a pair of data points from different classes that differ only in the values in S_1. If we assign the same value for S_1 then we introduce an inconsistency – two points that have the same attribute values but are classified in different classes. Therefore the last row of the matrix in Table 1 is a valid assignment for the intermediate concept.

The decomposition algorithm achieves higher generalization than the original data set. The construction of the sets defining the new functions g and h might lead to adding new points previously not present in the data set. This is due to the basic decomposition step at which some of the empty entries in the partition matrix might be assigned a value. This happens exactly when a non-empty entry exists in the same row which corresponds to the same value of the new concept. The generalization however does not necessarily extend to a cover of the whole input space. In the experiments in [10,11], a default rule was used which assigns the value of the most frequently used class in the example set that defines the intermediate concept.

A number of heuristics can be applied to limit the complexity, e.g. limit the size of S_1. Furthermore a number of partition selection measures have been investigated in [10,11], however, the most simple one, called column multiplicity, proved best. It chooses the partition with the least number of values of the new concept.

3 Decomposition with Monotonicity Constraints

Let us have a monotone data set D with an attribute set A and a discrete monotone labelling $\lambda : D \to \{0, m\}$. $S_0 \cap S_1 = \emptyset$, $S_0 \cup S_1 = A$. A scheme of the type $f = g(S_0, h(S_1))$ is called *positive* if the functions g, h are positive.

Monotone decomposition looks at the question whether for given S_0 and S_1 there exists an extension of the positive scheme $f = g(S_0, h(S_1))$. The requirement that $h(S_1)$ should be positive implies that the data set generated by S_1 and the corresponding values given by h should satisfy the monotonicity constraint. We denote this data set by $S_1|h$. Similarly the requirement that g should be positive implies that the resulting set (denoted by $S_0h|\lambda$) after applying h on S_1 in D should also satisfy the monotonicity constraint.

Let us use the example in Table 2. The attributes are split in subsets $S_0 = \{a_1, a_2, a_3\}$ and $S_1 = \{a_4, a_5, a_6\}$. We assume that g and h are known. Then the data set generated by applying h on S_1 is given in Table 3. This data set is required to be monotone. Using the values of h from Table 3 in the original table (in Table 2) we construct the set from Table 4. It is also required to be monotone.

The problem has so far been investigated in the context of *Boolean functions* in [5]. There a criterion is given for the existence of an extension of positive schemes of a number of different types. A partially defined Boolean function has an extension of positive scheme $f = g(S_0, h(S_1))$ if and only if there is no pair of vectors $x \in T^*$ and $y \in F^*$ such that $x[S_1] \leq y[S_1]$ (i.e. y dominates x on all attributes in the subset S_1). Here $x[S_1]$ denotes the vector containing only the values of x for the attributes in S_1 and T^*/F^* are defined as follows:

Table 2. A monotone data set

X	a_1	a_2	a_3	a_4	a_5	a_6	λ
x_1	3	2	2	1	3	1	3
x_2	2	2	2	2	3	1	3
x_3	3	1	3	1	3	2	3
x_4	2	1	1	1	1	2	2
x_5	2	2	3	1	2	1	2
x_6	1	1	2	2	2	1	2
x_7	1	1	2	1	1	1	1
x_8	1	2	1	1	1	1	1

Table 3. The new data set $S_1|h$

X	a_4	a_5	a_6	h
x_1	1	3	1	1
x_2	2	3	1	2
x_3	1	3	2	1
x_4	1	1	2	1
x_5	1	2	1	1
x_6	2	2	1	2
x_7	1	1	1	1
x_8	1	1	1	1

Table 4. The new data set $S_0h|\lambda$

X	a_1	a_2	a_3	h	λ
x_1	3	2	2	1	3
x_2	2	2	2	2	3
x_3	3	1	3	1	3
x_4	2	1	1	1	2
x_5	2	2	3	1	2
x_6	1	1	2	2	2
x_7	1	1	2	1	1
x_8	1	2	1	1	1

$$T^* = \{x \in T | \exists y \in F : y[S_0] \geq x[S_0]\}$$

$$F^* = \{y \in F | \exists x \in T : y[S_0] \geq x[S_0]\}.$$

Deciding if a partially defined Boolean function has an extension of such positive scheme can be done in polynomial time with complexity $O(m^2 n)$.

In this paper we investigate the corresponding problem in the context of *discrete functions*. First, the following lemma can be proven:

Lemma 1. *There exists a positive extension for the scheme $f = g(S_0, h(S_1))$ if and only if there exists an assignment of values $h : D \to \{h_i\}_{i=1}^k$ such that the two new data sets $S_1|h$ and $S_0h|\lambda$ are monotone.*

(see [6] for a proof)

Let us consider the data set generated by h, i.e. $S_1|h$. The monotonicity constraint here means that if $x[S_1] \leq y[S_1]$ then $h(x[S_1]) \leq h(y[S_1])$. We assign some unknown values to the class attribute $\{h_i\}_{i=0}^k$ and generate constraints of the type $h_i \leq h_j$ for the class values for each couple of different data points such that $x_i[S_1] \leq x_j[S_1]$. For the example of Table 2, the constraints generated in this way will be as shown in Table 5 where $h_i = h(x_i[S_1])$.

Table 5. The set of constraints from the data table $S_1|h$

$h_1 \geq h_5$	$h_2 \geq h_8$	$h_4 \geq h_8$
$h_1 \geq h_7$	$h_3 \geq h_1$	$h_5 \geq h_7$
$h_1 \geq h_8$	$h_3 \geq h_4$	$h_5 \geq h_8$
$h_2 \geq h_1$	$h_3 \geq h_5$	$h_6 \geq h_7$
$h_2 \geq h_5$	$h_3 \geq h_7$	$h_6 \geq h_8$
$h_2 \geq h_6$	$h_3 \geq h_8$	$h_7 \geq h_8$
$h_2 \geq h_7$	$h_4 \geq h_7$	$h_8 \geq h_7$

We now replace the vectors $x[S_1]$ with the corresponding (for the moment still unknown) values h_i and that results in the data set $S_0h|\lambda$. It should also be monotone which here means that if $x \leq y$ then x cannot belong to a higher class than y. We transform this constraint to a constraint over h: if $x[S_0] \leq y[S_0]$ and x belongs to a higher class than y then $h(x[S_1]) > h(y[S_1])$. For the example, the constraints of this type are: $h_2 > h_5$ and $h_6 > h_7$.

Note that if there are no constraints of the second type the remaining constraints of only the first type can be satisfied by assigning the same value to all h-variables and that would be a valid solution to the problem.

A natural way of representing the constraints is in a directed graph with vertices corresponding to $\{h_i\}_{i=1}^{k}$ and two types of directed edges. The first type of constraints (larger or equal) will be denoted by a dashed arrow: $x \rightarrow y$ will mean $x \geq y$. The second type (larger) will be represented by solid arrows where $x \rightarrow y$ will mean $x > y$. This representation will be used for finding a consistent assignment for h. Intuitively, such assignment cannot be found if a cycle is present such as: $x_1 \geq x_2$, $x_2 > x_3$, $x_3 \geq x_1$. The graph for our example is given in Figure 3.

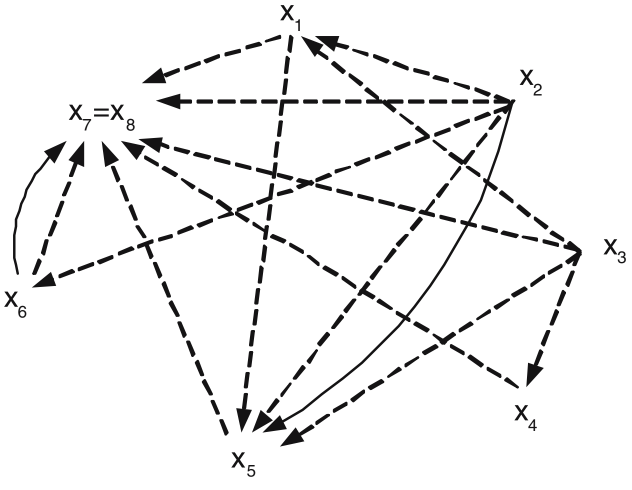


Fig. 3. The constraints graph for the example

3.1 Existence and Minimality of a Positive Extension of the Scheme $f = g(S_0, h(S_1))$

We can formulate a more precise criterion for whether there exists an assignment for the values $\{h_i\}_{i=1}^{k}$ consistent with all constraints. First we define the following notation:

$$T_i = \{x \in D : \lambda(x) = i\} \text{ for } i \in [0, m],$$

$$T_{>i} = \{x \in D : \lambda(x) > i\} \text{ for } i \in [0, m-1],$$

$$T_{<i} = \{x \in D : \lambda(x) < i\} \text{ for } i \in [1, m].$$

Using this, we define the following two sets:

$$T^{*}_{>i} = \{x \in T_{>i} \mid \exists y \in T_i, x[S_0] \leq y[S_0]\} \text{ for } i \in [0, m-1],$$

$$T^*_{<i} = \{x \in T_{<i} \,|\, \exists y \in T_i, y[S_0] \leq x[S_0]\} \text{ for } i \in [1, m].$$

Theorem 1. *There exists a positive extension of scheme $f = g(S_0, h(S_1))$ iff there are no data points $x_i, x'_i, x_{i+1}, x'_{i+1}, \ldots, x_{i+j}, x'_{i+j}, x_{i+j+1}$ such that $x_{i+j+1} = x_i$ and such that for all x_k, x'_k the following conditions hold:*

1. *if $x_k \in T_l$ then $x'_k \in T^*_{>l}$,*
2. *$x'_k[S_1] \leq x_{k+1}[S_1]$.*

(see [6] for a proof)

When an assignment exists it is hardly ever unique. In general we are interested in assignments with a (nearly) minimal number of values. In order to address this problem we first define a path in the constraint graph as a sequence of vertices $x_1, x_2, \ldots, x_j$ such that for each x_i, x_{i+1} there exists an edge from x_i to x_{i+1}(i.e. $h(x_i) < h(x_{i+1})$ or $h(x_i) \leq h(x_{i+1})$). The length of a path P is the number of solid edges in it, denoted by $|P|$. For an acyclic constraint graph, we denote the maximal length of a path in the graph by χ.

Theorem 2. *If there exists a positive extension of the scheme $f = g(S_0, h(S_1))$, then the minimal number of values necessary for an assignment consistent with the constraints equals the number $\chi + 1$ of the constraint graph.*

(see [6] for a proof)

Two possible assignments for h are (where $h_{\min}$ is the minimal possible value of h):

$$h_D(x) = \begin{cases} h_{\min} + \max\{|P| : P - \text{path starting from } x\} & \text{if such path exists,} \\ h_{\min} & \text{otherwise} \end{cases}$$

$$h'_D(x) = \begin{cases} h_{\min} + \chi - \max\{|P| : P - \text{path leading to } x\} & \text{if such path exists,} \\ h_{\min} + \chi & \text{otherwise.} \end{cases}$$

Lemma 2. *For each vertex x of an acyclic constraints graph, it holds that $h_D(x) \leq h'_D(x)$.*

(see [6] for a proof)

We apply a graph theory algorithm called topological sorting to order the vertices so that all edges point in the same direction (e.g. the graph of Fig. 3 is rearranged as in Fig. 4). We can now find an assignment as follows:

```
If no edges start from the vertex, assign $h_{\min}$;
Otherwise for each such edge:
    extract the label of the end vertex;
    for solid edges add 1 to the corresponding number;
    find the maximal among the numbers for all edges ending in $x$;
    assign this maximal number to the current vertex.
```

The complexity of the procedure is $O(|V||E|)$ where V is the set of vertices and E is the set of edges. For our example the new assignment is shown in Table 4.

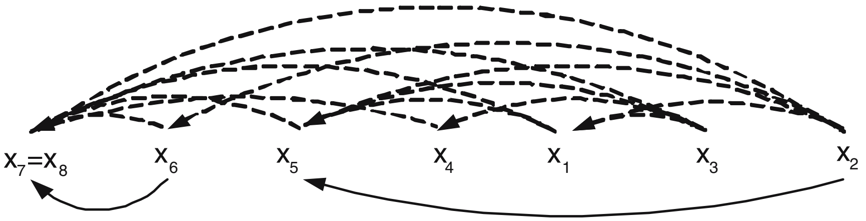


Fig. 4. The constraint graph from Figure 3 after the topological sorting

3.2 Default Rule for Covering the Whole Input Space

Our algorithm inherits the characteristic of the general decomposition algorithm that it cannot guarantee coverage of the whole input space. A default rule is needed that results in a monotone classifier. It should be applied at two steps - comparison with $S_1|h$ and with $S_0h|\lambda$. At the first step we compare $x[S_1]$ with the data set $S_1|h$. If none of the data points in it is equal to $x[S_1]$ we need to find a new label consistent with $S_1|h$. Then we replace $x[S_1]$ with it and compare the new example with the data set $S_0h|\lambda$. Here again if none of the examples is equal to the new one we have to find a new label that is consistent with the rest of the data.

Since these two steps are similar, we can use the same type of labelling function over the two data sets. We denote the data set by D which at the first step should be replaced by $S_1|h$ and at the second step by $S_0h|\lambda$. Similarly $\lambda(x)$ at the first step should be replaced by h and at the second step by g.

We propose two alternatives for the labelling function – $\lambda_{\min}$ and $\lambda_{\max}$, which are defined as follows:

$$\lambda_{\min}(x) = \begin{cases} \max\{\lambda(y): \ y \in D \cap \downarrow x\} & \text{if } x \in \uparrow D \\ c_{\min} & \text{otherwise} \end{cases}$$

$$\lambda_{\max}(x) = \begin{cases} \min\{\lambda(y): \ y \in D \cap \uparrow x\} & \text{if } x \in \downarrow D \\ c_{\max} & \text{otherwise} \end{cases}$$

where $\mathcal{X}$ is the input space, $c_{\min}/c_{\max}$ are the minimal and the maximal possible label respectively and we use the following other notation:

$$\downarrow x = \{y \in \mathcal{X} : y \leq x\},$$

$$\uparrow x = \{y \in \mathcal{X} : y \geq x\},$$

$$\downarrow D = \bigcup_{x \in D} \downarrow x,$$

$$\uparrow D = \bigcup_{x \in D} \uparrow x.$$

Different functions may be used on the different steps, e.g. $\lambda_{\min}$ for labelling at the first step and $\lambda_{\max}$ at the second step, as long as the same function is applied every time we are at the same step. If we apply both functions at the same step the monotonicity is no longer guaranteed. These functions were previously used in the Monotone Decision Trees algorithm ([2,7]) and are proven to give consistent labels when the data set is monotone. $\lambda_{\max}$ tends to give higher labels and more optimistic predictions than $\lambda_{\min}$.

As an example we take $x = (2, 2, 2, 2, 2, 2)$. $x[S_1] = (2, 2, 2)$ does not appear in $S_1|h$ (Table 3). We apply the labelling functions and both $\lambda_{\min}$ and $\lambda_{\max}$ give label 2. We then replace the label in x and compare the new data point $(2, 2, 2, 2)$ with $S_0h|\lambda$ (Table 4). It is not present there, therefore we need to apply again a labelling function. $\lambda_{\min}$ predicts a label 2 and $\lambda_{\max}$ predicts 3. Let us assume that we prefer more optimistic predictions and we choose $\lambda_{\max}$. Therefore the label assigned to $x = (2, 2, 2, 2, 2, 2)$ will be 3.

4 Experimental Results

In order to investigate the successfulness of the proposed algorithm, some experiments were conducted on the Nursery data set obtained from UCI ML Repository [3]. It is a real-world monotone data set of 12960 applications for a nursery school described by 8 attributes and covers the whole input space. It was generated using a hierarchical model developed by experts, see Figure 5, therefore has *known* underlying structure. The values of all attributes are ordered according to how inconvenient the situation is. Furthermore the underlying problem is monotone since the better the situation of the family is the more recommended it is to accept the application. We have previously used the same data set for experiments with another classification algorithm for monotone data - Monotone Decision Trees (see [1,6]).

In the experiments the following rules were used in order to handle tie situations. If one candidate is a subset of another candidate with the same score

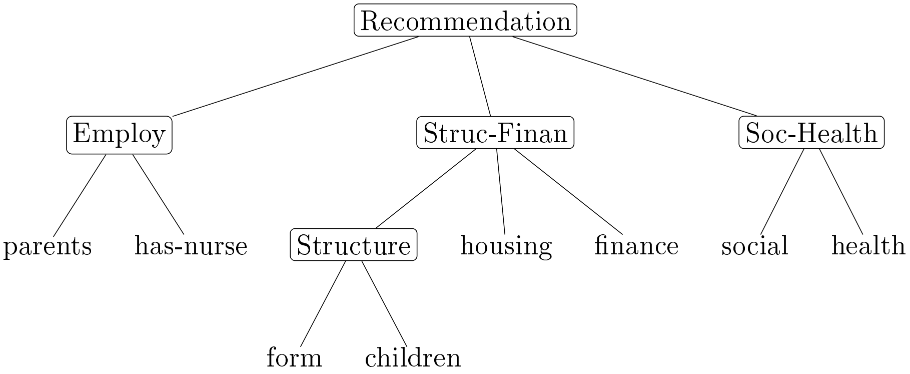


Fig. 5. The concept hierarchy used to generate the Nursery data set

then the superset candidate has priority. Right-breadth-first strategy is used in building the decomposition tree, i.e. among candidates with the same score, the right-most has priority and we first consider for splitting the remaining attributes on the same level before we look deeper. The minimal length of a candidate is 2, the maximal is fixed to 5. We used column multiplicity for partition selection and $h_D(x)$ for finding the new concept.

With these assumptions, the final decomposition structure extracted by the algorithm is identical to the one developed by the experts and used to produce the data set. It must be noted that with different design decisions (left-breadth-first/depth-first strategies) the algorithm takes a different path but generates equivalent structures. Further experiments were performed with a sample of 500 data points. The algorithm produced the same decomposition tree.

5 Conclusions

In this paper we propose a decomposition method for monotone discrete functions which generates a monotone classifier based on the extracted concept hierarchy. We formulate a criterion for the existence of a positive extension of the scheme $f = g(S_0, h(S_1))$ and propose a method for finding an assignment for the intermediate concept using minimal number of values. Furthermore we propose two monotone default rules for the classification of points not covered by the extended data set of the concept hierarchy.

At least two directions for further research can be mentioned. It would be interesting to investigate whether the results could be extended to cope with noise. It would also be interesting to consider decompositions with a restriction on the intermediate concepts.

References

1. Bioch, J.C. and Popova, V. (2002): Monotone Decision Trees and Noisy Data. In: H. Blockeel and M. Denecker (Eds.): *Proceedings of the 14th Belgium-Dutch Conference on Artificial Intelligence (BNAIC'2002).* Leuven, 19–26.
2. Bioch, J.C. and Potharst, R. (1997): Decision Trees for Monotone Classification. In: K. van Marcke and W. Daelmans (Eds.): *Proceedings of the Dutch Artificial Conference on Artificial Intelligence (NAIC'97)*, 361–369.
3. Blake, C.L. and Mertz, C.J. (1998): UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science [http://www.ics.uci.edu/~mlearn/ MLRepository.html].
4. Bohanec, M. and Rajkovič, V. (1990): DEX: An expert system shell for decision support. *Sistemica, 1, 145–157.*
5. Boros, E., Gurvich, V., Hammer, P.L., Ibaraki, T. and Kogan, A. (1995): Decomposability of partially defined Boolean functions. *Discrete Applied Mathematics, 62, 51-75.*
6. Popova, V. (2004) *Knowledge Discovery and Monotonicity*, PhD Thesis, Erasmus University Rotterdam, The Netherlands.
7. Potharst, R. and Bioch, J.C. (2000) Decision Trees for Ordinal Classification. *Intelligent Data Analysis, 4, 1–15.*

8. Samuel, A.L. (1959): Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development, 3, 221–229.*
9. Shapiro, A.D. (1987): *Structured induction in expert systems.* Turing Institute Press in association with Addison-Wesley, Wokingham, UK.
10. Zupan, B. (1997): *Machine learning by function decomposition.* PhD Thesis, University of Ljubljana.
11. Zupan, B., Bohanec, M., Demsar, J. and Bratko, I. (1999): Learning by Discovering Concept Hierarchies. *Artificial Intelligence, 109, 211–242.*

Learning On-Line Classification via Decorrelated LMS Algorithm: Application to Brain-Computer Interfaces

Shiliang Sun and Changshui Zhang

State Key Laboratory of Intelligent Technology and Systems,
Department of Automation, Tsinghua University, Beijing, China, 100084
sunsl02@mails.tsinghua.edu.cn, zcs@mail.tsinghua.edu.cn

Abstract. The classification of time-varying neurophysiological signals, e.g., electroencephalogram (EEG) signals, advances the requirement of adaptability for classifiers. In this paper we address the challenge of neurophysiological signal classification arising from brain-computer interface (BCI) applications and propose an on-line classifier designed via the decorrelated least mean square (LMS) algorithm. Based on a Bayesian classifier with Gaussian mixture models, we derive the general formulation of gradient descent algorithms under the criterion of LMS. Further, to accelerate convergence, the decorrelated gradient instead of the instantaneous gradient is adopted for updating the parameters of the classifier adaptively. Utilizing the presented classifier for the off-line analysis of practical classification tasks in brain-computer interface applications shows its effectiveness and robustness compared to the stochastic gradient descent classifier which uses the instantaneous gradient directly.

1 Introduction

Recently, the emerging research of brain-computer interface (BCI) technology, which is to give its users communication and control routes that do not depend on the brain's normal output channels of peripheral nerves and muscles, issues many challenges to the artificial intelligence community [1][2][3][4]. One of the big challenges in BCI applications is how to recognize the user's intent from the observation of neurophysiological signals as accurate as possible. In this paper, we focus on the classification problem of one particular variety of neurophysiological signals, namely electroencephalogram (EEG) signals which are electrical brain activities recorded from electrodes placed on the scalp.

Compared to magnetoencephalography (MEG), optical imaging, positron emission tomography (PET) and functional magnetic resonance imaging (fMRI), electroencephalography is a relatively inexpensive and convenient means to monitor the brain's activities. Although the recorded EEG signals suffer from the trouble of low signal noise rate (SNR), currently it is a rather recipient way (non-invasive and ethical) to access brain signals [5][6]. However, the essential nondeterminacy of brain activity implies the high variability of EEG recordings.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 215–226, 2005.

The EEG signals being used in a BCI are typically non-stationary, especially between two different sessions with a rather long time interval. Factors such as user's strategy, motivation, attention, fatigue or frustration may affect the features of EEG activities significantly. Besides, the environmental noise in all kinds of natural conditions can also cause the mental state to change by gradual degrees. As an instance, Millán had shown that two different mental tasks, imagination of left and right hand movements respectively, can have closer power maps than the same task during two consecutive sessions [7]. Altogether, the spontaneous variability of EEG recordings between experimental sessions makes it a difficult issue to categorize different EEG signals, and necessitates learning the on-line classification to boost up the performance of existing BCIs.

Hitherto, there is few work dealing with the problem of on-line learning for EEG signal classification in the literature. Although many on-line learning methods are available from the neural network, statistical, and computational learning disciplines, they are usually computationally expensive and do not suit BCI applications simply [8][9][10]. Our current work is initially inspired by several recent publications of Millán and his colleagues [7][11][12][13]. Although they presented to use the idea of stochastic gradient descent to carry out on-line learning of a statistical classifier, under their rather rigorous assumptions, they hadn't provided the formulations of variable updates in a systematic way. We will make up for this deficiency and discuss the related work later in the main text.

The main contribution of this article is that, based on a Bayesian classifier with Gaussian mixture models we derive the exact formulation of gradient algorithm in a much general way, and then present a decorrelated least mean square (DLMS) algorithm utilizing the theoretical outcome to learn the on-line classification of EEG signals in BCI applications. Real-world classification experiments with three kind of mental imagery tasks also verifies the effectiveness of our approach.

The remainder of this paper is organized as follows. Besides the theoretical derivation of gradient update, section 2 also covers the details of how to build up the on-line Bayesian classifier employing the idea of decorrelated LMS algorithm. Section 3 reports the experimental results for several BCI subjects on three mental imagery tasks. Then, in section 4 we discuss some related work. Finally, section 5 gives the conclusions and future work plan.

2 On-Line Classifiers

As we have stated before, the competence of on-line learning is very necessary in BCI applications. However, to the best of our knowledge, there is little work addressed this matter in the literature till now. The articles of Millán *et al.* are one of the first to bring forward this problem in the BCI settings [7][11][12][13]. For the on-line learning in BCIs, one would first encounter the problem of choose which kind of classifiers. For the consideration of low computation cost and practical superiority, here we adopt the Bayesian classifier to deal with the issue of multi-class categorization, as suggested by others [7][12].

2.1 Bayesian Classifier

Assume there are N samples in a training set which come from K categories, and each class denoted by C_k has prior $P(C_k)$, $(k = 1, ..., K)$, s.t., $\sum_{k=1}^{K} P(C_k) = 1$. For each class, its class conditional probability density is assumed to be the weighted combination of N_k Gaussian probability density functions, i.e.,

$$p(x|C_k) = \sum_{i=1}^{N_k} a_k^i G(x|\mu_k^i, \Sigma_k^i), s.t., \Sigma_{i=1}^{N_k} a_k^i = 1 \tag{1}$$

where $G(x|\mu_k^i, \Sigma_k^i)$ is a Gaussian probability density function with mean μ_k^i and covariance Σ_k^i [14]. According to Bayesian theorem [10], the posterior probability of x belonging to class C_k can be given as

$$\begin{aligned} P(C_k|x) &= \frac{P(C_k)p(x|C_k)}{p(x)} \\ &= \frac{P(C_k)\sum_{i=1}^{N_k} a_k^i G(x|\mu_k^i, \Sigma_k^i)}{\sum_{j=1}^{K} P(C_j)\sum_{i=1}^{N_j} a_j^i G(x|\mu_j^i, \Sigma_j^i)} \,. \end{aligned} \tag{2}$$

Now we represent the samples as $\{x_n, y_n\}, n = 1, ..., N$, whereas x_n is the feature vector, y_n is the corresponding label. If $x_n \in C_k$, then $y_n = e_k^K = \left[0, \ldots, 1_{(k)}, \ldots, 0\right]_{(K)}^{\top}$. Denote $\hat{y}_n$ as the outcome of our Bayesian classifier, i.e.,

$$\hat{y}_n = \left[P(C_1|x_n), P(C_2|x_n), \ldots, P(C_K|x_n)\right]^{\top} \,.$$

Under the criterion of least mean square (LMS), the cost function for unconstrained optimization becomes

$$\min J(\Theta) = \min E\{\|e_n\|^2\} = \min E\{\|y_n - \hat{y}_n\|^2\} \tag{3}$$

where variable Θ represents any of the parameters $N_k, a_k^i, \mu_k^i, \Sigma_k^i$. To make our analysis feasible, we only presume here that parameters N_k, a_k^i are given or obtained from previous training data, while parameters μ_k^i, Σ_k^i would have the most general form (μ_k^i is a general column vector, Σ_k^i is a symmetric and positive definite matrix) and would be updated through on-line learning.

For the application of LMS algorithm and the later mentioned decorrelated LMS algorithm, one should first derive the formulation of stochastic gradient (instantaneous gradient) $\nabla_{\Theta}\|y_n - \hat{y}_n\|^2$. Note that $\|y_n - \hat{y}_n\|^2$ can be rewritten as follows:

$$\begin{aligned} \|y_n - \hat{y}_n\|^2 &= (y_n - \hat{y}_n)^T (y_n - \hat{y}_n) \\ &= y_n^T y_n - 2y_n^T \hat{y}_n + \hat{y}_n^T \hat{y}_n \\ &= y_n^T y_n - 2\sum_{i=1}^{K} y_n^i P(C_i|x_n) + \sum_{j=1}^{K} (P(C_j|x_n))^2 \\ &= y_n^T y_n + \sum_{j=1}^{K} [(P(C_j|x_n))^2 - 2y_n^j P(C_j|x_n)] \,. \end{aligned} \tag{4}$$

Thus, we have

$$\nabla_{\Theta}\|y_n - \hat{y}_n\|^2 = 2\sum_{j=1}^{K}[(P(C_j|x_n) - y_n^j)\nabla_{\Theta}P(C_j|x_n)] \tag{5}$$

where Θ is μ_k^i or $(\Sigma_k^i)^{-1}$ (for computational convenience, we use $(\Sigma_k^i)^{-1}$ instead of Σ_k^i from now on) in this paper.

2.2 Derive $\nabla_{\mu_k^i} P(C_j|x_n)$

Define $\Phi_1 = \frac{P(C_k)a_k^i}{p(x_n)}G(x_n|\mu_k^i, \Sigma_k^i)(\Sigma_k^i)^{-1}(x_n - \mu_k^i)$, then

$$\nabla_{\mu_k^i}P(C_j|x_n) = \begin{cases} [1 - P(C_k|x_n)]\Phi_1, & for\ j = k \\ -P(C_j|x_n)\Phi_1, & for\ j \neq k \end{cases} \tag{6}$$

(see Appendix A for details).

2.3 Derive $\nabla_{(\Sigma_k^i)^{-1}} P(C_j|x_n)$

Because $\nabla_{\Sigma_k^i}P(C_j|x_n)$ is difficult to get directly, we try to derive $\nabla_{(\Sigma_k^i)^{-1}}P(C_j|x_n)$ alternatively.

$$\nabla_{(\Sigma_k^i)^{-1}}P(C_j|x_n) = \begin{cases} \frac{P(C_k)a_k^i}{p(x_n)}[1 - P(C_k|x_n)]\Phi_2, & for\ j = k \\ -\frac{P(C_k)a_k^i P(C_j|x_n)}{p(x_n)}\Phi_2, & for\ j \neq k \end{cases} \tag{7}$$

where

$$\Phi_2 = G(x_n|\mu_k^i, \Sigma_k^i)\{\Sigma_k^i - \frac{1}{2}diag(\Sigma_k^i) - A + \frac{1}{2}diag(A)\}$$

with $A = (x_n - \mu_k^i)(x_n - \mu_k^i)^{\top}$ (see Appendix B for details).

2.4 Decorrelated LMS Algorithm for Bayessian Classifier

With the derived stochastic gradient formulation in (5), one might seek to update parameter Θ using the gradient directly (namely LMS algorithm), i.e. using

$$\Theta_n = \Theta_{n-1} - \mu_n\nabla_{\Theta_{n-1}}\|y_n - \hat{y}_n\|^2 \tag{8}$$

to carry out on-line learning adaptively, where μ_n is the learning rate [15]. However, this would take a risk of low convergence rate and poor tracking performance, since stochastic gradient $\nabla_{\Theta_{n-1}}\|y_n - \hat{y}_n\|^2$ is only the instantaneous approximation of the true gradient which should be derived from $\nabla_{\Theta_{n-1}}E\{\|y_n - \hat{y}_n\|^2\}$. If two consecutive instantaneous gradients correlate with each other, then the mean square error (MSE) might be accumulated and couldn't be corrected in time. Therefore, to get rid of these shortcomings, here we adopt the decorrelated gradient instead of the instantaneous gradient [15][16]. Using decorrelated

Table 1. The flow chart of the decorrelated LMS (DLMS) algorithm for learning online classification

The variable Θ in the following procedure denotes μ_k^i or $(\Sigma_k^i)^{-1}$ with $\{k = 1, \ldots, K; i = 1, \ldots, N_k\}$.

Step 1:

Initialize Θ with Θ_0.

Step 2:

For $n = 1, 2, \ldots$, calculate the decorrelated gradient $\hat{\nabla}_{\Theta_{n-1}} \|y_n - \hat{y}_n\|^2$ from (5) and (9), and update Θ with $\Theta_n = \Theta_{n-1} - \mu_n \hat{\nabla}_{\Theta_{n-1}} \|y_n - \hat{y}_n\|^2$.

gradient can effectively avoid the case of error accumulation which might arise in instantaneous gradient descent algorithms, and hence, can accelerate the convergence of the adaptive-gradient methods.

The decorrelated gradient of Θ_n can be defined as

$$\hat{\nabla}_{\Theta_n} \|y_n - \hat{y}_n\|^2 = \nabla_{\Theta_n} \|y_n - \hat{y}_n\|^2 - a_n \nabla_{\Theta_{n-1}} \|y_n - \hat{y}_n\|^2 \tag{9}$$

where a_n is the decorrelation coefficient between $\nabla_{\Theta_n} \|y_n - \hat{y}_n\|^2$ and $\nabla_{\Theta_{n-1}} \|y_n - \hat{y}_n\|^2$. For two vectors v_n and v_{n-1}, the decorrelation coefficient a_n can be defined as

$$a_n = \frac{(v_n - \bar{v}_n)^\top (v_{n-1} - \bar{v}_{n-1})}{(v_{n-1} - \bar{v}_{n-1})^\top (v_{n-1} - \bar{v}_{n-1})} \tag{10}$$

where $\bar{v}_n$ represents the mean value of v_n [15]. For two matrices, the concept of decorrelation coefficient can be similarly extended. Table 1 describes the paradigm of our proposed decorrelated LMS (DLMS) algorithm for learning on-line classification.

3 Experiments

3.1 Materials and Protocols

Here we describe the data set analyzed in this paper. The data set contains EEG recordings from 3 normal subjects (denoted by A, B, C respectively) during nonfeedback mental imagery tasks. The subjects sat in a normal chair, relaxed arms resting on their legs. The three tasks are: imagination of repetitive self-paced left hand movements (class C_1), imagination of repetitive self-paced right hand movements (class C_2) and generation of different words beginning with the same random letter (class C_3).

For a given subject, there are 3 recording sessions acquired on the same day, each lasting about 4 minutes with breaks of 5-10 minutes in between. The subject performed a given task for about 15 seconds and then switched randomly to the next task at the operator's request. The raw EEG potentials were first spatially filtered by means of a surface Laplacian [17][18]. The superiority of surface Laplacian transformation over raw potentials for the operation of BCI has already been demonstrated [19]. Then, every 62.5 ms, the power spectral density in the band 8-30Hz was estimated over the last second of data with a frequency resolution of 2 Hz for 8 centro-parietal channels (EEG signals recorded over this region reflects the activities of brain's sensorimotor cortices). The power spectra in the frequency band 8-30 Hz were then normalized according to the total energy in that band. As a result, an EEG sample is a 96-dimensional vector (8 channels times 12 frequency components). The total number of samples for subjects A, B, and C during three sessions are respectively 3488/3472/3568, 3472/3456/3472, and 3424/3424/3440. For a more detailed description of the data and the brain computer interface protocol, please refer to [7]. In this article, we concentrate on utilizing the 96 dimensional pre-computed features to address the problem of on-line classification.

3.2 Experimental Results

EEG signal classification is conducted for each subject. First of all, to reduce the parameters to be estimated and avoid the over-fitting problem, principal component analysis (PCA) is adopted to reduce the feature dimensions by reserving 90% energy. The threshold 90% is a good tradeoff between dimension reduction and energy preservation for our problem. To initialize the parameters μ_k^i and Σ_k^i of the DLMS algorithm, we first apply the k-Means clustering algorithm with multiple runs [10], and the result with the least cost value is selected for initialization utility. On the selection of parameters $P(C_k), N_k$ and a_k^i in the Bayesian classifier of Gaussian mixture models, we take the same configuration as [7], for in his research, Millán had shown its effectiveness through cross-validations. Thus, $P(C_k) = \frac{1}{3}$, $N_k = 4$ and $a_k^i = \frac{1}{4}$ $(k = 1, 2, 3; i = 1, 2, 3, 4)$.

In this article, the data of session 1 from each mental task of every subject is employed to implement parameter initialization. For class C_k, we first use k-Means clustering algorithm to initialize μ_k^i which comes from one of the N_k cluster centers. Then Σ_k^i can be obtained using the data belonging to the same cluster C_k^i. Subsequently, we update the parameters adaptively on the first one minute data of the next session (the samples are processed sequentially and only once, to completely stimulate the on-line situation). With the final updated parameters, we test the performance of the classifier on the data of the last three minutes from the next session. The learning rate of μ_k^i and $(\Sigma_k^i)^{-1}$ are taken as 1e-6 and 1e-4 respectively, which are found to provide good classification results among a small number of parameter search for the basic LMS algorithm. The same procedure is performed on session 2 and session 3, i.e., we initialize the parameters μ_k^i and Σ_k^i through k-Means clustering on session 2, then update them using the first one minute data of session 3 and test the final classifier on the last three minute data of session 3.

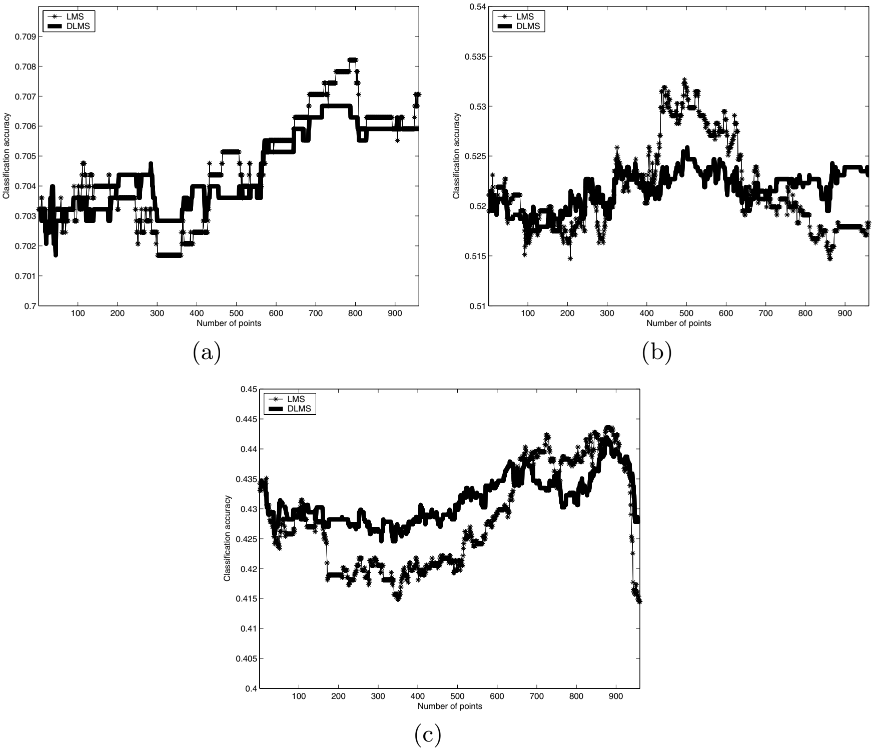
To evaluate the performance of our decorrelated LMS (DLMS) algorithm for learning on-line classification, under the same conditions we also carry out on-line classification using the basic LMS algorithm, which adopts instantaneous gradient instead of decorrelated gradient to update parameters. The final

Table 2. Classification accuracies of on-line learning by LMS algorithm and decorrelated LMS (DLMS) algorithm

Subjects	Sessions	LMS	DLMS
A	2	67.79%	67.87%
	3	70.71%	70.59%
B	2	47.40%	45.63%
	3	51.83%	52.31%
C	2	49.19%	48.78%
	3	41.45%	42.82%

classification accuracy rates using these two classifiers with parameters updated by the whole one minute data are given in Table 2.

Through statistical Z-test, no significant difference is found between the final results of these two algorithms (p-value=0.8845). This only indicates that the



(a)

(b)

(c)

Fig. 1. (a): The time course of classification accuracies on session 3, subject A. (b): The time course of classification accuracies on session 3, subject B. (c): The time course of classification accuracies on session 3, subject C.

Table 3. The standard deviations (STDs) (normalized to the range [1, 10]) of the time courses of on-line classification by LMS algorithm and DLMS algorithm

Subjects	Sessions	LMS STD	DLMS STD	STD Reduced
A	2	1.46	1.02	30.1%
	3	1.79	1.30	27.4%
B	2	9.95	2.22	77.7%
	3	4.32	1.87	56.7%
C	2	4.26	4.42	−3.8%
	3	8.39	3.89	53.6%
Average				40.28%

performance of LMS algorithm is statistically similar to DLMS algorithm after a long time of update. As we have stated before, one important requirement for on-line BCI applications is to improve the classification performance using as minimal training data as possible. Besides, for non-feedback BCIs, as there is no sign helping subjects to rectify their latent strategies generating EEG signals, effective algorithms should be of good stability. Below we give the time courses of the convergence of these two algorithms during the on-line update stage for classifying the last three minutes of session 3 of three subjects in Fig. 1. That is, after every update, we obtain the classification accuracy on the last 3 minutes of session 3. From Fig. 1, the robustness and the rapid convergence of DLMS algorithm are manifested. Although the results of LMS algorithm and DLMS algorithm have the same tendencies, the magnitude variance of classification accuracy obtained by DLMS algorithm is rather smaller than that of LMS algorithm. Thus the rapid convergence and robustness of DLMS algorithm are indicated. For other test sessions, similar results are observed. In addition, to give a quantitative description, the standard deviations of the classification results for LMS algorithm and DLMS algorithm are respectively given in Table 3, from which we can see that by using DLMS algorithm for gradient descent the standard deviation has been reduced to a large extent.

4 Related Work

With regard to the idea of stochastic gradient descent, Millán *et al.,* have mentioned it in their publications [7][11][13]. However, they usually make a very rigorous assumption about the formulation of covariance matrix Σ_k^i, such as the assumption of diagonal and common to all the prototypes of a certain class, and make a simple approximation about the gradient of μ_k^i. Fig. 2 shows the distribution of two features from the original 96 ones. Clearly, using the combination of diagonal covariances could not represent the external oblique distribution logically.

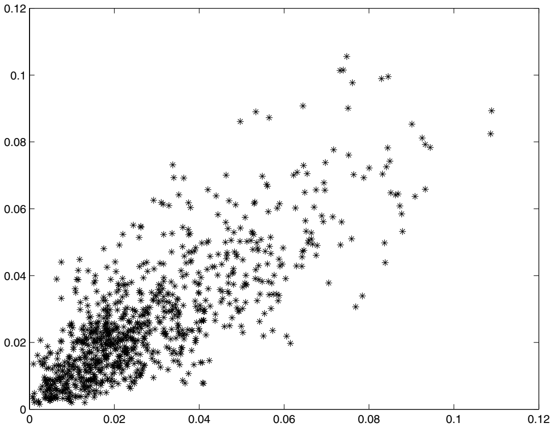


Fig. 2. The distribution of two features from the original 96 ones of session 1, subject A

While in this article, PCA is adopted for dimension reduction and the covariance matrices are described with a general form. This would be more reasonable and more powerful in depicting different data distributions. Besides, we independently derive the general representation of gradient descent algorithm for μ_k^i and $(\Sigma_k^i)^{-1}$ in a Bayesian classifier context, which didn't appear before in the literature as far as we know. In addition, a new algorithm namely decorrelated LMS algorithm is proposed for the on-line learning of μ_k^i and $(\Sigma_k^i)^{-1}$, and obtains better performance than the basic LMS algorithm (stochastic gradient descent algorithm). These make our current work much different from Millán's.

5 Conclusions and Future Work

The research of brain-computer interface technology is an interdisciplinary project, which gestates many challenges in a variety of aspects. In this paper, we address the problem of on-line classification of EEG signals with applications to brain-computer interfaces. The time-varying characteristic of EEG recordings between experimental sessions makes it a difficult issue to categorize different EEG signals, and necessitates learning the on-line classification. Based on a Bayesian classifier of Gaussian mixture models, we derive the general formulations of the instantaneous gradient and the decorrelated gradient. Besides, a decorrelated LMS algorithm (DLMS) is developed to accelerate the convergence of the traditional LMS algorithm (stochastic gradient descent method). Experiments and comparisons shows the effectiveness and robustness of our approach.

For practical utilities, one can design a easy-going protocol to implement on-line learning. Each time users make use of BCI equipments after a long break, there would be a on-line learning stage of one minute or so during which a display device generates a series of random signs indicating upcoming tasks. Following these cues, users carry out specific mental activities. Simultaneously, the classifier would be updated on-line. In the future, study on the realization of automatic on-line training and on the active selection of training instances would be an interesting issue.

Acknowledgements

Shiliang Sun and Changshui Zhang would like to thank IDIAP Research Institute (Switzerland) for providing the analyzed data. Besides, we would also like to thank the National Natural Science Foundation of China for supporting this work under Project 60475001.

References

1. Nicolelis, M.A.L.: Actions from Thoughts. Nature, Vol. 409 (2001) 403-407
2. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-Computer Interfaces for Communication and Control. Clinical Neurophysiology, Vol. 113 (2002) 767-791
3. Ebrahimi, T., Vesin, J.M., Garcia, G.: Brain-Computer Interfaces in Multimedia Communication. IEEE Signal Processing Magazine, Vol. 20 (2003) 14-24
4. Wickelgren, I.: Tapping the Mind. Science, 299 (2003) 496-499
5. Wolpaw, J.R., Birbaumer, N., Heetderks, W.J., McFarland, D.J., Peckham, P.H., Schalk, G., Donchin E., Quatrano, L.A., Robinson, C.J., Vaughan, T.M.: Brain-Computer Interface Technology: A Review of the First International Meeting. IEEE Transactions on Rehabilitation Engineering, Vol. 8 (2000) 164-173
6. Vaughan, T.M.: Guest Editorial Brain-Computer Interface Technology: A Review of the Second International Meeting. IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 11 (2003) 94-109
7. Millán, J.R.: On the Need for On-Line Learning in Brain-Computer Interfaces. Proceedings of 2004 International Joint Conference on Neural Networks. Budapest, Hungary (2004)
8. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
9. Saad, D.: On-Line Learning in Neural Networks. Cambridge University Press (1998)
10. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. 2th edn. John Wiley & Sons, New York (2000)
11. Millán, J.R., Renkens, F., Mouriño, J., Gerstner, W.: Non-Invasive Brain-Actuated Control of a Mobile Robot. Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, (2003) 1121-1126
12. Millán, J.R., Renkens, F., Mouriño, J., Gerstner, W.: Brain-Actuated Interaction. Artificial Intelligence, Vol. 159 (2004) 241-259
13. Millán, J.R., Renkens, F., Mouriño, J., Gerstner, W.: Noninvasive Brain-Actuated Control of a Mobile Robot by Human EEG. IEEE Transactions on Biomedical Engineering, Vol. 51 (2004) 1026-1033
14. Mclachlan, G., Peel, D.: Finite Mixture Models. Wiley, New York (2000)
15. Glentis, G.O., Berberidis, K., Theodoridis, S.: Efficient Least Square Adaptive Algorithms for FIR Transversal Filtering. IEEE Signal Processing Magzine, Vol. 16 (1999), 13-41

16. Doherty, J., Porayath, R.: A Robust Echo Canceler for Acoustic Environments. IEEE Transactions on Circuits and Systems, II, Vol, 44 (1997) 389-398
17. Perrin, R., Pernier, J., Bertrand, O., Echallier, J.: Spherical Spline for Potential and Current Density Mapping. Electroencephalography and Clinical Neurophysiology, Vol. 72 (1989), 184-187
18. Perrin, R., Pernier, J., Bertrand, O., Echallier, J.: Corrigendum EEG 02274. Electroencephalography and Clinical Neurophysiology, Vol. 76 (1990), 565
19. McFarland, D.J., McCane, L.M., David, S.V., Wolpaw, J.R.: Spatial Filter Selection for EEG-Based Communication. Electroencephalography and Clinical Neurophysiology, Vol. 103 (1997) 386-394

Appendix A: Derive $\nabla_{\mu_k^i} P(C_j|x_n)$

$$\begin{aligned}\nabla_{\mu_k^i} P(C_j|x_n) &= \nabla_{\mu_k^i} \frac{P(C_j)p(x_n|C_j)}{p(x_n)} \\ &= \nabla_{\mu_k^i} \frac{P(C_j)\sum_{l=1}^{N_j} a_j^l G(x_n|\mu_j^l, \Sigma_j^l)}{p(x_n)} \qquad (11)\end{aligned}$$

A.1 When $j = k$

$$\begin{aligned}&\nabla_{\mu_k^i} \frac{P(C_j)\sum_{l=1}^{N_j} a_j^l G(x_n|\mu_j^l, \Sigma_j^l)}{p(x_n)} \\ &= \frac{P(C_k)a_k^i}{p(x_n)}[1 - P(C_k|x_n)]\nabla_{\mu_k^i} G(x_n|\mu_k^i, \Sigma_k^i) \qquad (12)\end{aligned}$$

where

$$\nabla_{\mu_k^i} G(x_n|\mu_k^i, \Sigma_k^i) = G(x_n|\mu_k^i, \Sigma_k^i)(\Sigma_k^i)^{-1}(x_n - \mu_k^i) \ . \qquad (13)$$

A.2 When $j \neq k$

$$\begin{aligned}&\nabla_{\mu_k^i} \frac{P(C_j)\sum_{l=1}^{N_j} a_j^l G(x_n|\mu_j^l, \Sigma_j^l)}{p(x_n)} \\ &= -\frac{P(C_k)a_k^i P(C_j|x_n)}{p(x_n)}\nabla_{\mu_k^i} G(x_n|\mu_k^i, \Sigma_k^i) \ . \qquad (14)\end{aligned}$$

Appendix B: Derive $\nabla_{(\Sigma_k^i)^{-1}} P(C_j|x_n)$

$$\begin{aligned}\nabla_{(\Sigma_k^i)^{-1}} P(C_j|x_n) &= \nabla_{(\Sigma_k^i)^{-1}} \frac{P(C_j)p(x_n|C_j)}{p(x_n)} \\ &= \nabla_{(\Sigma_k^i)^{-1}} \frac{P(C_j)\sum_{l=1}^{N_j} a_j^l G(x_n|\mu_j^l, \Sigma_j^l)}{p(x_n)} \end{aligned} \tag{15}$$

B.1 When $j = k$

$$\begin{aligned}&\nabla_{(\Sigma_k^i)^{-1}} \frac{P(C_j)\sum_{l=1}^{N_j} a_j^l G(x_n|\mu_j^l, \Sigma_j^l)}{p(x_n)} \\ &= \frac{P(C_k)a_k^i}{p(x_n)}[1 - P(C_k|x_n)]\nabla_{(\Sigma_k^i)^{-1}} G(x_n|\mu_k^i, \Sigma_k^i)\ .\end{aligned} \tag{16}$$

Considering the normal distribution $G(x_n|\mu_k^i, \Sigma_k^i) = \frac{1}{(2\pi)^{d/2}|\Sigma_k^i|^{1/2}} exp\{-\frac{1}{2}(x_n - \mu_k^i)^\top (\Sigma_k^i)^{-1}(x_n - \mu_k^i)\} = \frac{1}{(2\pi)^{d/2}|\Sigma_k^i|^{1/2}} exp\{-\frac{1}{2}tr[(\Sigma_k^i)^{-1}(x_n - \mu_k^i)(x_n - \mu_k^i)^\top]\}$, if we denote $A = (x_n - \mu_k^i)(x_n - \mu_k^i)^T$, then

$$\begin{aligned}&\nabla_{(\Sigma_k^i)^{-1}} G(x_n|\mu_k^i, \Sigma_k^i) \\ &= \frac{1}{(2\pi)^{d/2}} exp\{-\frac{1}{2}tr[(\Sigma_k^i)^{-1}A]\}\frac{1}{2}|(\Sigma_k^i)^{-1}|^{-\frac{1}{2}}|(\Sigma_k^i)^{-1}|[2\Sigma_k^i - diag(\Sigma_k^i)] + \\ &\quad G(x_n|\mu_k^i, \Sigma_k^i)\{-\frac{1}{2}[2A - diag(A)]\} \\ &= G(x_n|\mu_k^i, \Sigma_k^i)\{\Sigma_k^i - \frac{1}{2}diag(\Sigma_k^i) - A + \frac{1}{2}diag(A)\}.\end{aligned} \tag{17}$$

B.2 When $j \neq k$

$$\begin{aligned}&\nabla_{(\Sigma_k^i)^{-1}} \frac{P(C_j)\sum_{l=1}^{N_j} a_j^l G(x_n|\mu_j^l, \Sigma_j^l)}{p(x_n)} \\ &= -\frac{P(C_k)a_k^i P(C_j|x_n)}{p(x_n)}\nabla_{(\Sigma_k^i)^{-1}} G(x_n|\mu_k^i, \Sigma_k^i).\end{aligned} \tag{18}$$

An Algorithm for Mining Implicit Itemset Pairs Based on Differences of Correlations

Tsuyoshi Taniguchi and Makoto Haraguchi

Division of Computer Science, Hokkaido University,
N-14 W-9, Sapporo 060-0814, Japan
{tsuyoshi, makoto}@kb.ist.hokudai.ac.jp

Abstract. Given a transaction database as a global set of transactions and its local database obtained by some conditioning to the global one, we consider a pair of itemsets whose degrees of correlations are higher in the local database than in the global one. A problem of finding paired itemsets with high correlation in one database is known as Discovery of Correlation, and some algorithms to search for such characteristic paired itemsets are already proposed. However, even non-characteristic paired itemsets in the local database are also meaningful, provided the degree of correlation increases much higher in the local database than in the global one. They can be an implicit and hidden evidence showing that something particular to the local database occurs even though they are not yet realized as characteristic ones in the local. From this viewpoint, we have already proposed to measure the significance of paired itemsets by the difference of two correlations before and after the conditioning to the local database, and define a notion of DC pairs whose degrees of differences of correlations are high. As DC pairs are regarded as compound itemsets consisting of two component itemsets, we can have two basic strategies for finding them. One strategy firstly examines the compound itemsets and then the components, while another one does the component itemsets and then the compound ones. According to the former strategy, which we have already proposed and tested for its effectiveness, we have to enumerate many number of candidate compound itemsets that cannot be decomposable to components. For this reason, this paper presents a new algorithm according to the second strategy. It firstly enumerate possible component itemsets based on a new pruning rule for cutting off useless components. Secondly it forms the compound itemsets by combining the components thus detected, while we also make use of a constraint for preventing our algorithm from checking meaningless combinations.

1 Introduction

In the studies of data mining from transaction databases, many studies have been paying much attention to finding itemsets with high supports, paired itemsets appeared in association rules with high confidence [4], or paired itemsets with strong correlation [8,9,10,11]. These notions are considered useful for distinguishing characteristic paired itemsets with strong correlation in a single transaction

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 227–240, 2005.

database. A similar strategy based on the notion of change of supports, known as Emerging Patterns [5], is successful even for finding itemsets characterizing either of two databases. All of these notions about itemsets are thus proposed to extract paired itemsets required to be characteristic in a given database or either of two or more databases.

Although some users regard characteristic paired itemsets with strong correlation as useful, others may often regard many number of such paired itemsets as trivial because of the reason that they have been already known without examining a database. On the other hand, as is indicated in the study of Chance Discovery [12], some itemsets not characteristic in the above sense are also useful, as they are *potentially significant* under some condition.

For instance, suppose a database in which information about ages of customers and goods they purchased are stored. There may exist several pairs of particular ages and goods with high correlations, if people at those ages have a general tendency to buy those goods. In this case, the degree of correlation is not much dependent on time stamp data. As a result, there will be a little difference between the degree of correlation in the whole database and one in a local database of transactions with the recent time stamp data. On the other hand, there may exist another kinds of goods which teen-agers, for instance, begin to drastically buy just recently. As the purchase actions made by those aged people just begin, the overall degree of correlation between the ages and the goods is still low. However, its degree observed by restricting the transactions to those with the recent time stamp will show a significantly higher value.

Thus, the notion of potential significance we would like to define is the difference of degrees of correlation before and after some conditioning by which a local database is derived. Although we can consider various ways of conditioning and the corresponding local databases, we try to present a general algorithm to find a significant paired itemset with high change ratio of correlations, given a global and a local database. In section 6, a database with items designating places of transactions is examined. In this case, the conditioning is given by specifying particular place of transactions. The task of our algorithm is to find paired itemsets with higher correlation in the particular area, compared with the correlation in the whole area. Again it should be noted that the former correlation in the particular area need not be high, as we can interpret such a paired itemset as an implicit evidence showing that something particular to the local area occurs.

From the viewpoints mentioned in the above, we have already defined the notion of DC pairs and presented an algorithm to find them in [1]. More precisely, given a global and a local transaction databases, an itemset pair with higher change ratio of correlations is called a DC pair. A DC pair is syntactically regarded as a compound itemset consisting of two component itemsets. So, the algorithm presented in [1] is designed so that it firstly examines the compound itemsets and then the components, using two parameters for restricting the search spaces for the compound and the component itemsets. Although the algorithm is equipped with some pruning rules, an experimental result showed that large number of useless compound itemsets never decomposable into candidates in the space of component itemsets are generated and tested. Consequently, the subspace our algorithm actually visited turned out to be a very large one.

From the experimental observation thus obtained, in this paper, we present another new algorithm that enumerates component itemsets firstly and then combines those detected components into compound ones. It is clear from the definition that there exists no chance for the algorithm to examine any compound itemset not decomposable to possible component itemsets. Additionally, we can show that it suffices to check only transactions containing the candidates components in the local database in order to identify possible combinations of components. As the number of such transactions is not many, our algorithm can effectively generate compound itemsets from the set of candidate component itemsets. On the other hand, in the process of generating components, we can enjoy a monotone property over itemsets, depending on the parameters, that is also useful to prevent our algorithm from generating useless component itemsets.

Thus, in both processes of generating components and of combining them into compound itemsets, the number of generated candidates are restricted.

2 Related Works and Paper Organization

There exist many works in the field of data mining that are based on a strategy of contrasting two or more databases in order to extract significant properties or patterns from a huge data set. Particularly, data mining techniques, known as contrast-set mining [5,6,7], have been designed specifically to identify differences between databases to be contrasted.

For instance, in the study of Emerging Patterns [5] for two transaction databases, itemsets whose supports are significantly higher in one database than in another one are considered significant, as they can be candidate patterns for distinguishing the former from the latter. A similar strategy is also used in the system STUCCO [6] in order to obtain characteristic itemsets in one database based on χ^2 test. In addition, the system, Magnum Opus [7], examines relations between itemsets and a database among several databases. On the other hand, what this paper tries to find are paired itemsets whose correlations drastically increase in one database. Thus we can say that the subject of this paper is a kind of "contrast-set mining of correlations between itemsets".

Secondly, many methodologies have been proposed to detect characteristic correlations in a single database [8,9,10]. In these studies, using some function measuring the degree of correlation between itemsets, strongly correlated itemsets in a given database or in one database from given two databases are examined. Thus, these methods are also used to discover itemsets or family of itemsets that are characteristic in one database. On the other hand, the algorithm presented in this paper is designed so as to find even paired itemsets whose correlation in one database is not significantly high but is significantly higher than correlation in another database. Our algorithm may find the characteristic paired itemsets as special cases, but is never supposed to find only characteristic ones. To find these paired itemsets, we present some new pruning rules so that the algorithm successfully detects even non-characteristic paired itemsets.

Several notions about correlations have been proposed and used in the above previous studies from information theoretic or statistical viewpoints. If we need to consider even negative events $\overline{Y}$ that an itemset Y does not appear in transactions, the notion of correlations between two itemsets X and Y based on χ^2-test shall be taken into account. However, this paper is concerned with the notion of correlation in the sense that the number of chances for Y to occur increases under the presence of X. The degree of correlation in this sense can be calculated by the ratio $P(Y|X)/P(Y)$, known as self-mutual information by taking log.

Finally, we discuss the relation between a condition itemset which decide a local database and itemset pairs we try to find. If the condition is regarded as an antecedent of some rule, the itemset pairs can be considered a consequent of the rule. For example, association rule [4] is a rule whose consequent is an itemset such that the conditional probability of the itemset given by the antecedent (the condition in this paper) is no more than some parameter. That is, itemsets such that the probability of the itemsets is high in local database can be found by association rule. On the other hand, we find even itemsets such that the probability of the itemsets is very low in local database as a result of detecting high changes of correlation by the conditioning to the local database. Further, the correlation between the condition and the itemset pair is not always high. Briefly speaking, we try to find rules such that a consequent of the rule is an implicit itemset pair with high degree of change of correlations under an antecedent (a condition) of the rule.

The rest of this paper is organized as follows. The next section defines some terminologies used throughout this paper. In Section 4, we introduce the notion of DC pairs and define our problem of mining DC pairs. An algorithm for finding DC pairs is described in Section 5. Section 6 presents our experimental results. In the final section, we summarize our study and discuss future work.

3 Preliminaries

Let $\mathcal{I} = \{i_1, i_2, \cdots, i_n\}$ be a set of *items*. An *itemset* is a subset of $\mathcal{I}$. A *transaction database* $\mathcal{D}$ is a set of transactions, where a transaction is an itemset. We say that a transaction t *contains* an itemset X, if $X \subseteq t$. For a transaction database $\mathcal{D}$ and an itemset X, the *occurrence* of X over $\mathcal{D}$, denoted by $O(X, \mathcal{D})$, is defined as $O(X, \mathcal{D}) = \{t | t \in \mathcal{D} \wedge X \subseteq t\}$, and the *probability* of X over $\mathcal{D}$, denoted by $P(X)$, is defined as $P(X) = |O(X, \mathcal{D})|/|\mathcal{D}|$.

For an itemset C, a *sub-database* of $\mathcal{D}$ w.r.t. C, denoted by $\mathcal{D}_C$, is defined as the set of transactions containing C in $\mathcal{D}$, that is, $\mathcal{D}_C = O(C, \mathcal{D})$. The *complement* of $\mathcal{D}_C$ w.r.t. $\mathcal{D}$ is denoted by $\overline{\mathcal{D}_C}$ and is defined as $\overline{\mathcal{D}_C} = \mathcal{D} - \mathcal{D}_C$.

For itemsets X and Y, the *correlation* between X and Y over a transaction database $\mathcal{D}$, $correl(X, Y)$, is defined as $correl(X, Y) = P(X \cup Y)/P(X)P(Y)$. For a sub-database $\mathcal{D}_C$, the correlation between X and Y over $\mathcal{D}_C$, $correl_C(X, Y)$, is given by $correl_C(X, Y) = P(X \cup Y|C)/P(X|C)P(Y|C)$, where $P(X|C) = P(X \cup C)/P(C)$. Note here that correlations are defined for only itemsets X whose supports in $\mathcal{D}$ and $\mathcal{D}_C$ are non-zero. We regard a pair of X and Y such that $correl(X,Y) > 1$ as characteristic since $P(X|Y) > P(X)$ holds. Note that $P(Y|X) > P(Y)$ holds, too. Similarly, we regard a pair of X and Y such that $correl(X,Y) \leq 1$ as non-characteristic.

4 DC Pair Mining Problem

In this section, we define a notion of DC pairs and our problem of mining them.

For a pair of itemsets X and Y, we especially focus on "difference of correlations observed by conditioning to a local database". Suppose here that an itemset C is a condition given by users. The difference of correlations is measured by the following ratio:

$$change(X,Y;C) = \frac{correl_C(X,Y)}{correl(X,Y)} = \frac{P(C)P(C|X \cup Y)}{P(C|X)P(C|Y)}. \tag{1}$$

Let $\rho(> 1)$ be an admissible degree of difference of correlations. In our framework, a pair of itemsets X and Y is considered significant if $change(X,Y;C) \geq \rho$ holds. Since we assume C is given by users, $P(C)$ can be regarded as a constant. Therefore, the difference is actually evaluated with the following function g:

$$g(X,Y;C) = \frac{P(C|X \cup Y)}{P(C|X)P(C|Y)}. \tag{2}$$

A pair of itemsets X and Y is called a *DC pair* if $g(X,Y;C) \geq \rho/P(C)$. We try to find all DC pairs efficiently. It should be noted here that the function g behaves *non-monotonically* according to expansion of itemsets X and Y. So we cannot apply a simple pruning method like one Apriori adopted [4]. Therefore, we approximate the above problem according to the following naive strategy:

> Find pairs of X and Y which give higher values of $P(C|X \cup Y)$, keeping the values of $P(C|X)$ and $P(C|Y)$ small.

In order to control the values of $P(C|X \cup Y)$, we use a new parameter ζ $(0 \leq \zeta \leq 1)$. Given ρ and ζ, we use a new parameter ϵ such that $\epsilon^2 = \zeta \cdot P(C)/\rho$ in order to control each value of $P(C|X)$ and $P(C|Y)$. Note here that ϵ can be replaced with another parameter if the value of $P(C|X)P(C|Y)$ is low.

Definition 1. DC Pairs Mining Problem
Let C be an itemset for conditioning. Given ρ and ζ, the DC pair mining problem is to find any pairs of X and Y such that $P(C|X \cup Y) > \zeta$, $P(C|X) < \epsilon$ and $P(C|Y) < \epsilon$, where $\epsilon = \sqrt{\zeta \cdot P(C)/\rho}$. We say that $X \cup Y$ is a *compound itemset* and X and Y are *component itemsets.*

5 An Algorithm for Finding DC Pairs

In this section, we present an algorithm to solve the DC pair mining problem. At first, we discuss a basic strategy of finding DC pairs. Next, we prove a pruning rule in order to find candidates for component itemsets efficiently. Finally, we show some constraints of DC pairs in order to restrict the combinations of the candidates properly.

5.1 A Basic Strategy of Finding DC Pairs

At first, we discuss a basic strategy of finding DC pairs. The DC pairs we try to find are pairs of itemsets X and Y such that $X \cup Y$ is a compound itemset and X and Y are component itemsets. Then, two strategies of finding DC pairs can be considered mainly. One strategy is that compound itemsets are identified and each compound itemset is divided into component itemsets. And another strategy is that component itemsets are identified and their compound itemsets are found. The former strategy has already been tried to find DC pairs in [1] and there exist some difficulties. In order to explain the difficulties, we show properties of $P(C|X)$ and DC pairs based on the following observation.

Consider an itemset X appeared in a global and a local databases. And note that $P(X) \neq 0$ and $P(C \cup X) \neq 0$ must hold. Also, as the size of X is longer, $P(X)$ and $P(C \cup X)$ tend to become lower. Since $P(C|X)$ is non-zero, $P(C|X)$ tend to be high in the case that $P(X)$ is low. That is, when the size of X is long, $P(C|X)$ tends to be high. This means that there exist a few candidates for component itemsets whose size is long in the database. Next, consider two itemsets X and Y and suppose here that the size of $X \cup Y$ is almost the same size of maximal transactions in the database. Since the size of either X or Y is necessarily no more than the half size of $X \cup Y$, either $P(C|X)$ or $P(C|Y)$ tend to be high. This means that there are a small number of DC pairs X and Y such that the size of $X \cup Y$ is long because either $P(C|X) < \epsilon$ or $P(C|Y) < \epsilon$ is difficult to hold. Therefore, there are many candidates for compound itemsets which cannot be divided into DC pairs in the database.

Based on the above observation, there is a difficulty of the strategy of finding candidates for compound itemsets. So, in this paper, we discuss the strategy of finding candidates for component itemsets in a bottom-up manner. Of course, if the number of the candidates components is large, the number of combinations of the candidate components is also very large. If the number of the candidates is N, the number of the combinations is $O(N^2)$. However, as the combinations can be restricted by using some constraint, it is expected that the number of the combinations is not so many.

After all, our strategy of finding DC pairs is that component itemsets are identified firstly, and the computation for mining DC pairs is divided into two phases:

Phase1: Identifying Component Itemsets
An itemset X such that $P(C|X) < \epsilon$ is identified as a candidate for a component itemset.

Phase2: Combining Component Itemsets
One component X is combined with another one Y such that $P(C|X \cup Y) > \zeta$.

5.2 Pruning Search Branches in Phase 1

In Section 4, by using parameters ζ and ϵ, we restrict DC pairs we try to find. Although $P(C|X)$ behaves *non-monotonically* according to expansion of an itemset X as well as g, we prove that a monotone property over itemsets can be observed depending on ϵ. In Phase 1, we consider a problem of mining candidates for component itemsets X in a bottom-up manner. During this search, we can prune useless branches (itemsets) based on the following observation.

Let X be an itemset and Z be an itemset containing X. Suppose that there exists a superset Z' of X such that $X \subseteq Z' \subseteq Z$ and $P(C|Z') < \epsilon$. Since $P(C|Z') = P(C)P(Z'|C)/P(Z') < \epsilon$, $P(C \cup Z') < \epsilon \cdot P(Z')$ holds. Therefore, $P(C \cup Z) \leq P(C \cup Z') < \epsilon \cdot P(Z') \leq \epsilon \cdot P(X)$. As the result, we have $P(C \cup Z) < \epsilon \cdot P(X)$. This means that if $P(C \cup Z) \geq \epsilon \cdot P(X)$ holds, then we cannot obtain any Z' such that $P(C|Z') < \epsilon$. That is, if $P(C \cup Z) \geq \epsilon \cdot P(X)$ holds, any Z' does not have to be examined.

Pruning Rule:
For a search node (itemset) X and a superset Z such that $X \subseteq Z' \subseteq Z$, if $P(C \cup Z) \geq \epsilon \cdot P(X)$, any Z' never be a candidate node of X in our search process.

When X examined in our bottom-up search can be applied to the above pruning rule, Z' do not have to be examined. However, there is a problem that the way of identifying a superset Z of X properly. We describe the way of identifying Z and a termination condition in Phase 1 in the next section.

5.3 Termination Condition in Phase1

In the previous section, we present a pruning rule in Phase 1. In order to use our pruning rule effectively, in sub-database $\mathcal{D}_{\mathcal{C}}$, while an itemset X is examined in a bottom-up manner, a superset Z of X have to be checked simultaneously. And, if the size of Z is long and our pruning rule can be applied to, many itemsets can be pruned. In order to identify such Z, the notion of look ahead in [2] can be used. Originally, the notion is used to find a frequent maximal itemset by checking a superset of an itemset examined. In order to use the notion, suppose a lexical ordering of the items and let X be an itemset examined at present. Let $\mathit{tail}(X)$ be the greatest item of X and $T(\mathit{tail}(X))$ be a set of a possible item which is greater than $\mathit{tail}(X)$ according to the lexical ordering. Then, X is expanded by adding an item $i \in T(\mathit{tail}(X))$ in order to avoid duplications. Note here that an itemset Z such that $Z = X \cup T(\mathit{tail}(X))$ is a potentially frequent maximal itemset which does not contain other itemsets in the database. That is, the size of Z is approximately the size of a maximal transaction and Z whose size is long is useful for pruning of many search nodes (itemsets). Further, although Z is not always a maximal itemset, we do not have to check whether Z is a maximal itemset or not. Rather, Z in this case is also useful for our search because a maximal itemset is not always able to be applied to the pruning rule and an itemset whose size is middle may be applied to. It should be noted that the cost of checking Z is not so high as only $\mathcal{D}_{\mathcal{C}}$ is examined.

Termination Condition of search in Phase 1:
For an itemset X and an itemset $Z = X \cup T(tail(X))$, if $P(C \cup Z) \geq \epsilon \cdot P(X)$, X is not expanded further in our search.

5.4 An Algorithm for Finding Candidates for Component Itemsets

We show a termination condition of our search in Phase 1 in the previous section. In this section, in order to implement the termination condition, we simply explain an algorithm for finding candidates for component itemsets.

At first, we use *backtracking* algorithm [2,3] in order to enumerate candidates for component itemsets. Backtracking algorithm is based on recursive calls. Normally, an iteration of the algorithm inputs a frequent itemset F whose probability is no more than some parameter, and generates itemsets by adding every possible items to F. However, an iteration of our algorithm inputs an itemset X whose probability is non-zero because even itemsets whose probability is very low may be DC pairs and it is difficult to set a really proper parameter of probability. For each itemset whose probability is non-zero among itemsets generated, the iteration generates recursive calls with respect to it. To avoid duplications, an iteration of backtracking algorithms adds items contained $T(tail(X))$.

Next, when an itemset X is examined, a proper $T(tail(X))$ is not known yet. Let i be $tail(X)$, $\mathcal{D}$ be a global database and $\mathcal{D}_{\mathcal{C}}$ be its local database. Then, the probability of $X \cup \{j\}$ $(j \in T(tail(X - \{i\})), j > i)$ in $\mathcal{D}$ and $\mathcal{D}_{\mathcal{C}}$, and $X \cup T(tail(X - \{i\}))$ in $\mathcal{D}_{\mathcal{C}}$ have to be calculated. The probability of $X \cup T(tail(X - \{i\}))$ is calculated in order to check whether X fulfill the termination condition or not. On the other hand, the probability of $X \cup \{j\}$ is calculated in order to check whether j can be contained $T(tail(X))$ or not. That is, although we do not use any parameter of probability, an itemset whose probability is zero is, of course, trivial. Because the probability of a superset of $X \cup \{j\}$ is zero if the probability of $X \cup \{j\}$ is zero, j do not have to be added $T(tail(X))$. In order to calculate the probability of $X \cup \{j\}$ efficiently, the notion of *occurrence deliver* [3] can be used. Let $\{j_1, j_2, \cdots, j_m\}$ be $T(tail(X - \{i\}))$. Occurrence deliver computes the probability of $X \cup \{j_1\}, X \cup \{j_2\}, \cdots, X \cup \{j_m\}$ at once by tracing transactions containing X in $\mathcal{D}$ and $\mathcal{D}_{\mathcal{C}}$. It uses a bucket for $j_1, j_2, \cdots, j_m$, and set them to empty set at the beginning. Then, for each transaction t containing X, occurrence deliver inserts t to the bucket of $j_1, j_2, \cdots, j_m$. After these insertions, the bucket of $j_1, j_2, \cdots, j_m$ is equal to $O(X \cup \{j_1\}, \mathcal{D}_{\mathcal{C}}), O(X \cup \{j_2\}, \mathcal{D}_{\mathcal{C}}), \cdots, O(X \cup \{j_m\}, \mathcal{D}_{\mathcal{C}})$ if $\mathcal{D}_{\mathcal{C}}$ is examined.

Based on the above techniques, our algorithm for finding candidates for component itemsets is summarized as follows. In the algorithm, suppose that, for each item e such that $P(C \cup \{e\}) \neq 0$, $P(C \cup \{e\})$ and $P(\{e\})$ are calculated in advance. And let $T(tail(\{e\} - tail(\{e\})))$ be $T(tail(\{e\}))$. For each item e, by using the following algorithm, the candidates for component itemsets can be found.

```
ALGORITHM FindCandidateComponent(X)
IF $P(C \cup X)/P(X) < \epsilon$ then Output $X$;
$T(tail(X)) = \emptyset$;    $count\_look\_ahead = 0$;
For each item $i$ such that $i \in T(tail(X - \{tail(X)\}))$ do
   $Bucket_c[i] = \emptyset$;    $Bucket[i] = \emptyset$;
End for
For each transaction $t_c$ such that $t_c \in \mathcal{D}_\mathcal{C}$ and $X \subseteq t_c$ do
   IF $(C \cup X \cup T(tail(X - \{tail(X)\}))) \subseteq t_c$ then $count\_look\_ahead{+}{+}$;
                                                        // look ahead
   For each item $j$ such that $j \in t_c$ and $j > tail(X)$ do
      Insert $t_c$ to $Bucket_c[j]$;                     // occurrence deliver
      IF $j \notin T(tail(X))$ then $T(tail(X)) = T(tail(X)) \cup \{j\}$;
   End for
End for
IF $T(tail(X)) \neq \emptyset$ and $P(C \cup X \cup T(tail(X - \{tail(X)\}))) < \epsilon \cdot P(X)$
then                                                     // our pruning rule
   For each transaction $t$ such that $t \in \overline{\mathcal{D}_\mathcal{C}}$ and $X \subseteq t$ do
                                                        //$\overline{\mathcal{D}_\mathcal{C}} = \mathcal{D} - \mathcal{D}_C$
      For each item $k$ such that $k \in t$ and $k \in T(tail(X))$ do
         Insert $t$ to $Bucket[k]$;                      // occurrence deliver
      End for
   End for
   For each item $e$ such that $e \in T(tail(X))$
      $O(X \cup \{e\}, \mathcal{D}_\mathcal{C}) = Bucket_c[e]$;
      $O(X \cup \{e\}, \mathcal{D}) = Bucket_c[e] + Bucket[e]$;
      IF $P(C \cup X \cup \{e\}) \neq 0$ then call FindCandidatePart($X \cup \{e\}$);
   End for
End if
```

5.5 Constraints of DC pairs in Phase2

After we find candidates for component itemsets in Phase 1, we have to combine one component with another one in order to find DC pairs finally. If the number of the candidates is large, the number of the combinations is very large. However, two constraints of DC pairs can be used in order to restrict the combination.

At first, we describe a basic constraint of DC pairs. The DC pairs are pairs of itemsets X and Y such that two itemsets do not overlap. Then, if both X and Y contain some same item, pairs of X and Y are not DC pairs. In this case, combined itemsets $X \cup Y$ do not have to be examined. Secondly, we explain a main constraint of DC pairs. If pairs of X and Y are DC pairs, $P(C \cup X \cup Y) \neq 0$ must hold. Therefore, Y is necessarily contained by transactions containing X in a sub-database $\mathcal{D}_\mathcal{C}$. Also, $P(C \cup X)$ is low in many case if $P(C|X) < \epsilon$ holds. For example, if $\epsilon = 0.1$ and $P(X) = 0.5$, $P(C \cup X) < 0.05$. Therefore, we firstly check whether or not Y is contained by transactions which contain X in $\mathcal{D}_\mathcal{C}$, and if Y is not contained by the transaction, Y does not have to be examined.

The combinations actually examined in detail are restricted properly by only checking the small number of such transactions.

6 An Experiment

In this section, we present our experimental results. The purpose of experiments is to confirm that DC pairs can be found efficiently by using our pruning rules and constraints, and potentially significant DC pairs can be actually found for a given database.

6.1 Dataset and Implementation

We conducted the experiments on *Entree Chicago Recommendation Data*, a database in the UCI KDD Archive [13]. It consists of eight databases each of which contains restaurant features in a region, e.g. Atlanta, Boston and so on in the USA. The eight databases are combined into a single database $\mathcal{D}$ referred to as the global one. With the conditioning by each region C, we define a local (sub-)database $\mathcal{D}_C$ in $\mathcal{D}$. The global database consists of 4160 transactions each of which is a subset of 265 items, where each item represents a feature of restaurant, e.g., "Italian", "romantic", "parking" and so on. Thus, a transaction $\{f_1, f_2, f_3\}$ means there exists a restaurant with the feature f_1, f_2 and f_3.

Based on the algorithm presented in the previous section, our system has been implemented in C and run on a PC with 1.00 GB RAM and a Xeon 3.60 GHz processor.

6.2 An Effect of our Pruning Rule

In this section, we show an effect of our pruning rule in Phase 1, that is, in the search for finding the candidates for component itemsets. Our experimental result is summarized in Figure 1. In the figure, N is the number of possible itemsets with the probability of non-zero in the local database we are concerned with. That is, it is the size of the whole search space. The computation time for extracting the candidates without the pruning is denoted by t_N. N_{act} is the number of itemsets actually examined in our search with the pruning and $t_{N_{act}}$ is the computation time for the search. N_{cand} denotes the number of the extracted candidates for component itemsets.

The result shows that the number of the candidates to be extracted, N_{cand}, is much smaller than the number of the possible ones in each case (region). Therefore, finding the candidates without any pruning will be quite impractical. As shown in the figure, since the pruning rule can reduce at least 90 % of the whole search space, it can be considered that our pruning can work well to improve the search efficiency in Phase 1.

6.3 An Effect of Constraints of DC Pairs

Let $Comp_{cand}$ be the set of the candidates obtained in Phase 1. In Phase 2, we examine whether a pair of component itemsets in $Comp_{cand}$ can be a DC pair

$\rho = 3.0, \zeta = 0.4$							
region	$P(C)$	ϵ	N	N_{act}	N_{cand}	t_N(sec)	$t_{N_{act}}(sec)$
Atlanta	0.0642	0.0922	2.4×10^7	1.8×10^6	3.5×10^4	144.031	17.906
Boston	0.105	0.118	2.4×10^8	4.5×10^6	4.5×10^4	1428.641	43.985
Chicago	0.163	0.147	4.8×10^7	3.1×10^6	4.7×10^4	283.172	28.735
Los Angeles	0.108	0.118	2.7×10^7	1.8×10^6	1.2×10^4	161.578	17.656
New Orleans	0.0786	0.102	1.5×10^7	1.4×10^6	1.2×10^4	89.656	12.735
New York	0.289	0.196	1.6×10^7	1.5×10^6	7.2×10^4	95.078	14.578
San Francisco	0.0995	0.114	3.0×10^7	2.3×10^6	5.3×10^4	176.141	21.750
Washington DC	0.0940	0.112	2.0×10^9	2.9×10^7	2.2×10^4	11536.375	279.500

Fig. 1. An effect of our pruning rule

region	$\lvert C_{all}\rvert$	$\lvert C_b\rvert$	$\lvert C_m\rvert$	$\lvert DC\rvert$	$\lvert DC_{imp}\rvert$	$t_{\lvert C_b\rvert}$(sec)	$t_{\lvert C_m\rvert}(sec)$
Atlanta	6.1×10^8	2.8×10^8	3.0×10^6	1.4×10^6	353	355.922	132.422
Boston	1.0×10^9	4.5×10^8	4.5×10^6	2.9×10^6	240	804.329	223.016
Chicago	1.1×10^9	5.8×10^8	4.5×10^6	3.1×10^6	7	829.906	236.282
Los Angeles	6.7×10^7	4.1×10^7	5.4×10^5	2.5×10^5	101	54.062	16.375
New Orleans	7.4×10^7	4.5×10^7	5.3×10^5	2.2×10^5	57	57.656	20.062
New York	2.6×10^9	1.2×10^9	9.7×10^6	6.8×10^6	44	2820.750	551.547
San Francisco	1.4×10^9	6.0×10^8	4.2×10^6	2.5×10^6	393	900.907	285.953
Washington DC	2.4×10^8	1.3×10^8	9.1×10^5	6.0×10^5	86	216.579	59.079

Fig. 2. An effect of constraint of DC pairs in Phase 2

or not. The results for Phase 2 is summarized in Figure 2. In the figure, $|C_{all}|$ is the number of the possible pairs we can extract from $Comp_{cand}$.

$|C_b|$ is the number of pairs of X and Y in $Comp_{cand}$ such that $X \cap Y = \phi$ and $t_{|C_b|}$ is the computation time for finding the DC pairs from C_b. Furthermore, $|C_m|$ is the number of pairs of X and Y in $Comp_{cand}$ such that both of X and Y are contained in a transaction in $\mathcal{D}_{\mathcal{C}}$. The computation time for finding the DC pairs from C_m is denoted by $t_{|C_m|}$. Finally, $|DC|$ is the number of extracted DC pairs and $|DC_{imp}|$ the number of DC pairs in DC whose degree of correlation is less than or equal to 1.

From the results, by the latter constraint, the number of candidate pairs to be examined can be drastically reduced. Therefore, it is expected that our search in Phase 2 can be performed efficiently. As the result, it is shown that DC pairs can be found efficiently by using our pruning rule and constraint. Further, in the next section, we show our search for DC pairs in this paper is efficient in contrast with our previous search in [1].

6.4 A Comparison Our New Method with Our Previous Method

As we discussed in 5.1, we have already tested the way of finding DC pairs that compound itemsets are firstly found in [1]. In order to compare our new method

region	N	N_{comp}	N_{DC}	$N_{comp} - N_{DC}$	$\|C_m\| - \|DC\|$
Atlanta	2.42×10^7	2.36×10^7	4.69×10^5	2.32×10^7	1.64×10^6
Boston	2.42×10^8	2.42×10^8	1.01×10^6	2.41×10^8	1.56×10^6
Chicago	4.78×10^7	4.76×10^7	5.41×10^5	4.70×10^7	1.44×10^6
Los Angeles	2.74×10^7	2.72×10^7	7.56×10^4	2.72×10^7	2.80×10^5
New Orleans	1.51×10^7	1.49×10^7	8.12×10^4	1.49×10^7	3.08×10^5
New York	1.59×10^7	1.56×10^7	8.83×10^5	1.47×10^7	2.95×10^6
San Francisco	2.96×10^7	2.88×10^7	7.90×10^5	2.80×10^7	1.69×10^6
Washington DC	1.95×10^9	1.95×10^9	2.76×10^5	1.95×10^9	3.06×10^5

Fig. 3. a comparison our new method with our previous method

with our previous one, we examine the number of candidates for compound itemsets and whether the candidates can be divided into DC pairs or not. Our experimental result is summarized in Figure 3. In the figure, N is the same number in 6.2, and $|DC|$ and $|C_m|$ are the same number in 6.3. N_{comp} is the number of candidates for compound itemsets in each region. N_{DC} is the number of the candidates which can be divided into DC pairs. Note here that the candidate may be several DC pairs. Although N_{DC} differs from $|DC|$ in 6.3, as a result, the same number of DC pairs can be found by using our previous method.

The experimental result shows that the number of candidates for compound itemsets, N_{comp}, is almost the same number of possible itemsets with the probability of non-zero in the local database, N, in each region. That is, there are a few itemsets which can be pruned even if some pruning rules presented in [1] are used. Therefore, in Phase 1, our new method can be found candidates efficiently in contrast with our previous method.

Although it is difficult to realize efficient search in Phase 1 by using our previous method, this does not mean that the previous method is not efficient if most of the candidates can be divided into DC pairs. However, as we discussed in 5.1, the number of the candidates which can be divided into DC pairs, N_{DC}, is much smaller than N_{comp}, so most of the candidates cannot be divided into DC pairs. Therefore, we have to check many candidates which cannot be DC pairs, further, may examine all subsets of each the candidates in worst case. On the other hand, the number of combinations of candidates for component itemsets which cannot be DC pairs, $|C_m| - |DC|$, is much smaller than the number of the candidates for compound itemsets which cannot be DC pairs, $N_{comp} - N_{DC}$. As a result, by using our new method, we do not have to examine many combinations which cannot be DC pairs.

Thus, it can be considered that our new method realize efficient mining of DC pairs in contrast with our previous method.

6.5 An Example of DC Pair

We have obtained various kinds of DC pairs in the experimental data. For instance, in New Orleans, a DC pair $X = \{$*Entertainment*, *Quirky*, *Up and Coming*$\}$ and $Y = \{$*\$ 15-\$ 30*, *Private Parties*, *Spanish*$\}$ has been found. The pair shows high degree of difference of correlations by conditioning to New Orleans. However, since the pair shows very high degree of correlation in the local database, we will be able to find them as characteristic itemsets by some method previously proposed. On the other hand, we have also found a DC pair, $X = \{\textit{Quirky}\}$ and $Y = \{\textit{Good Decor}, \textit{Italian}, \textit{\$15-\$30}, \textit{Good Service}\}$ for New Orleans. The pair is not correlated in both global database and local database. Therefore, the pair cannot be found by previous methods. However, the pair shows high degree of difference of correlations by conditioning to New Orleans. Although the correlation of the pair in New Orleans seems to be not so high, it is much higher than one in the global database. We consider such a DC pair can be especially useful in some cases. For instance, people looking for a restaurant in New Orleans may be interested in a "quirky Italian restaurant" which is a hidden feature in New Orleans in contrast with a "quirky Spanish restaurant" which is an explicit feature in the local database because there may be some factor of its high degrees of difference of correlations even if the pair does not show high degree of correlation.

Thus, our algorithm has actually found potentially significant DC pairs for the given database.

7 Concluding Remarks

Given a transaction database $\mathcal{D}$ and its sub-database $\mathcal{D}_{\mathcal{C}}$, we proposed the notion of DC pairs. A pair of itemsets X and Y is called a DC pair if the correlation between X and Y in $\mathcal{D}_{\mathcal{C}}$ is relatively high to one in the original $\mathcal{D}$ with some degree. It should be noted that the correlation is not always high in $\mathcal{D}_{\mathcal{C}}$ even though we can observe some degree of correlation change for $\mathcal{D}$ and $\mathcal{D}_{\mathcal{C}}$. In this sense, such a pair might not be characteristic in $\mathcal{D}_{\mathcal{C}}$. Thus, some DC pairs are regarded as *potential characteristics* in $\mathcal{D}_{\mathcal{C}}$. Our experimental results showed that DC pairs which are potentially significant can be actually found for "Entree Chicago Recommendation Data" under conditioning by each region.

In order to efficiently find DC pairs, we investigated several pruning mechanisms which can prune useless search nodes (branches) and designed an algorithm adopted them. The computation is divided into two phases. In Phase 1, we can efficiently extract the set of candidates for component itemsets with a look ahead strategy. In Phase 2, then, a restricted pairs of obtained candidates are examined whether they can be DC pairs or not. Our experimental results have also shown effectiveness of the pruning rules in our search.

A more powerful pruning mechanism would be desired in more practical cases. We would be able to realize such an improvement of computational efficiency heuristically. For instance, imposing a *semantic constraint* on itemsets will be effective in reducing our search space. We might consider only candidates for compound itemsets each of which contains a certain pair of items semantically interesting. As the result, the number of candidates can be drastically reduced still preserving semantical significance. This kind of constraints will be investigated as future work.

References

1. T. Taniguchi, M. Haraguchi and Y. Okubo. Discovery of hidden correlations in a local transaction database based on differences of correlations. In *Proc. of the IAPR Int'l Conf. on Machine Learning and Data Mining in Pattern Recognition*, 2005 (to appear).
2. R. J. Bayardo. Efficiently mining long patterns from databases. In *Proc. of the ACM-SIGMOD Int'l Conf. on Management of Data, 85-93*, 1998.
3. T. Uno, M. Kiyomi and H. Arimura. LCM ver.2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets. In *Proc. of the IEEE Int'l Conf. on data mining, 2nd Workshop on Frequent Itemset Mining Implementations (FIMI'04)*, 2004.
4. R. Agrawal, R. Srikant. Fast algorithms for mining association rules. In *Proc. of the Int'l Conf. on Very Large Data Bases*, pages 487-99, 1994.
5. G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proc. of the 5th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 43-52, 1999.
6. S. D. Bay and M. J. Pazzani. Detecting group differences: mining contrast sets. *Data Mining and Knowledge Discovery*, v 5, n 3, pages 213-46, 2001.
7. G. I. Webb, S. Butler and D. Newlands. On detecting differences between groups. In *Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 256-65, 2003.
8. S. Brin, R. Motwani and C. Silverstein. Beyond market baskets: generalizing association rules to correlations. In *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, v 26, n 2, pages 265-76, 1997.
9. S. Brin, R. Motwani, J. D. Ullman and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, v 26, n 2, pages 255-64, 1997.
10. C. C. Aggarwal, P. S. Yu. A new framework for itemset generation. In *Proc. of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1998)*, pages 18-24, 1998.
11. S. Morishita and J. Sese. Traversing itemset lattices with statistical metric pruning. In *Proc. of the ACM SIGACT-SIGMOD-SIGART Symposium on Database Systems (PODS).* pages 226-36, 2000.
12. Y. Ohsawa and Y. Nara. Understanding internet users on double helical model of chance-discovery process. In *Proc. of the IEEE Int'l Symposium on Intelligent Control*, pages 844-9, 2002.
13. S. Hettich, and S. D. Bay, The UCI KDD Archive, Department of Information and Computer Science, University of California, Irvine, CA, http://kdd.ics.uci.edu, 1999.

Pattern Classification via Single Spheres

Jigang Wang, Predrag Neskovic, and Leon N. Cooper*

Institute for Brain and Neural Systems,
Department of Physics,
Providence, RI 02912 USA
{jigang, pedja, Leon_Cooper}@brown.edu
http://physics.brown.edu/physics/researchpages/Ibns/index.html

Abstract. Previous sphere-based classification algorithms usually need a number of spheres in order to achieve good classification performance. In this paper, inspired by the support vector machines for classification and the support vector data description method, we present a new method for constructing single spheres that separate data with the maximum separation ratio. In contrast to previous methods that construct spheres in the input space, the new method constructs separating spheres in the feature space induced by the kernel. As a consequence, the new method is able to construct a single sphere in the feature space to separate patterns that would otherwise be inseparable when using a sphere in the input space. In addition, by adjusting the ratio of the radius of the sphere to the separation margin, it can provide a series of solutions ranging from spherical to linear decision boundaries, effectively encompassing both the support vector machines for classification and the support vector data description method. Experimental results show that the new method performs well on both artificial and real-world datasets.

1 Introduction

When objects are represented as d-dimensional vectors in some input space, classification amounts to partitioning the input space into different regions and assigning unseen objects in those regions into their corresponding classes. In the past, people have used a wide variety of shapes, including rectangles, spheres, and convex hulls, to partition the input space.

Spherical classifiers were first introduced into pattern classification by Cooper in 1962 and subsequently studied by many other researchers [1,2,3,4]. One well known classification algorithm consisting of spheres is the Restricted Coulomb Energy (RCE) network. The RCE network, first proposed by Reilly, Cooper, and Elbaum, is a supervised learning algorithm that learns pattern categories by representing each class as a set of prototype regions - usually spheres [5,6]. The RCE network incrementally creates spheres around training examples that are not covered, and it adaptively adjusts the sizes of spheres so that they do

* This work is partially supported by ARO under grant DAAD19-01-1-0754. Jigang Wang is supported by a dissertation fellowship from Brown University.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 241–252, 2005.

not contain training examples from different classes. After the training process, only the set of class-specific spheres is retained and a new pattern is classified based on which sphere it falls into and the class affiliation of that sphere.

Another learning algorithm that is also based on spherical classifiers is the set covering machine (SCM) proposed by Marchand and Shawe-Taylor [7]. In their approach, the final classifier is a conjunction or disjunction of a set of spherical classifiers, where every spherical classifier dichotomizes the whole input space into two different classes with a sphere. The set covering machine, in its simplest form, aims to find a conjunction or disjunction of a minimum number of spherical classifiers such that it classifies the training examples perfectly.

Regardless of whether the influence of a sphere is local (as in the RCE network) or global (as in the SCM), classification algorithms that use spheres normally need a number of spheres in order to achieve good classification performance, and therefore have to deal with difficult theoretical and practical issues such as how many spheres are needed and how to determine the centers and radii of the spheres. In this paper, inspired by the support vector machines (SVMs) for classification [8,9,10] and the support vector data description (SVDD) method [11,12], we propose a new method, which computes a single sphere that separates data from different classes with the maximum separation ratio. In contrast to previous methods that construct spheres in the input space, the proposed method constructs the separating sphere in the feature space induced by the kernel. Because the class of spherical boundaries in the feature space actually represents a much larger class than in the input space, our method is able to construct a single sphere in the feature space that separates patterns that would otherwise be inseparable when using a sphere in the input space.

Furthermore, when the ratio of the radius of the separating sphere to the separation margin is small, a sphere is constructed that gives a compact description of one class, coinciding with the solution of the SVDD method; and when the ratio is large, the solution effectively coincides with the maximum margin hyperplane solution. Therefore, by adjusting the ratio, the new method effectively encompasses both the support vector machines for classification and the SVDD method for data description, and may lead to better generalization performance than both methods.

The remainder of the paper is organized as follows. In Section 2 we give a brief overview of the support vector data description method that computes a minimum enclosing sphere to describe a set of data from a single class. In Section 3, we propose our new algorithm, which extends the SVDD method by computing a single sphere that separates data from different classes with the maximum separation ratio. In Section 4 we test the new algorithm on both artificial and real-world datasets. Concluding remarks are given in Section 5.

2 Support Vector Data Description

The basic idea of the SVDD method is to construct a minimum bounding sphere to describe a set of given data. The minimum bounding sphere, which is defined as the smallest sphere enclosing all data, was first used by Schölkopf, Burges, and Vapnik to estimate the VC-dimension of support vector classifiers and later applied by Tax and Duin to data description [11,12].

Given a set of training data $x_1, \ldots, x_n \in \mathbb{R}^d$, the minimum bounding sphere S, characterized by its center c and radius R, can be found by solving the following constrained quadratic optimization problem

$$\min_{c,R} \; R^2 \;\; , \tag{1}$$

subject to the constraints

$$\|x_i - c\|^2 \leq R^2 \quad \forall i = 1, \ldots, n \;\; . \tag{2}$$

To allow for the possibility of some examples falling outside of the sphere, one can relax the constraints (2) with a set of soft constraints:

$$\|x_i - c\|^2 \leq R^2 + \xi_i \quad \forall i = 1, \ldots, n \;\; , \tag{3}$$

where $\xi_i \geq 0$ are slack variables introduced to allow some examples to have larger distances. To penalize large distances to the center of the sphere, one can therefore minimize the following quadratic objective function

$$\min_{c,R,\xi_i} \; R^2 + C \sum_{i=1}^{n} \xi_i \;\; , \tag{4}$$

under the constraints (3), where $C > 0$ is a constant that controls the trade-off between the size of the sphere and the number of examples that possibly fall outside of the sphere.

Using the Lagrange multiplier method, the constrained quadratic optimization problem can be formulated as the following Wolfe dual form

$$\min_{\alpha_i} \sum_{i,j} \alpha_i \alpha_j \langle x_i, x_j \rangle - \sum_{i} \alpha_i \langle x_i, x_i \rangle \tag{5}$$

subject to the constraints

$$\sum_{i=1}^{n} \alpha_i = 1 \quad \text{and} \quad 0 \leq \alpha_i \leq C \quad \forall i = 1, \ldots, n \;\; . \tag{6}$$

Solving the dual quadratic programming problem, one obtains the Lagrange multipliers α_i for all $i = 1, \ldots, n$, which give the center c of S as a linear combination of x_i

$$c = \sum_{i=1}^{n} \alpha_i x_i \;\; . \tag{7}$$

According to the Karush-Kuhn-Tucker (KKT) optimality conditions, we have

$$\begin{aligned} \alpha_i = 0 \Rightarrow \|x_i - c\|^2 < R^2 &\quad \text{and} \quad \xi_i = 0 \\ 0 < \alpha_i < C \Rightarrow \|x_i - c\|^2 = R^2 &\quad \text{and} \quad \xi_i = 0 \\ \alpha_i = C \Rightarrow \|x_i - c\|^2 \geq R^2 &\quad \text{and} \quad \xi_i \geq 0 \;\; . \end{aligned}$$

Therefore, only α_i that correspond to training examples x_i which lie either on or outside of the sphere are non-zero. All the remaining α_i are zero and the corresponding training examples are irrelevant to the final solution. Knowing c, one can subsequently determine the radius R from the KKT conditions by letting

$$R^2 = \langle x_i, x_i \rangle - 2 \sum_{j=1}^{n} \alpha_i \langle x_i, x_j \rangle + \sum_{j,l} \alpha_j \alpha_l \langle x_j, x_l \rangle \tag{8}$$

for any i such that $0 < \alpha_i < C$.

In practice, training data of a class is rarely distributed spherically, even if the outermost examples are excluded. To allow for more flexible descriptions of a class, one can apply the kernel trick by replacing the inner products $\langle x_i, x_j \rangle$ in the dual problem with suitable kernel functions $k(x_i, x_j)$. As a consequence, training vectors x_i in $\mathbb{R}^d$ are implicitly mapped to feature vectors $\Phi(x_i)$ in some high dimensional feature space $\mathbb{F}$ such that inner products in $\mathbb{F}$ are defined as $\langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j)$, and spheres are constructed in the feature space $\mathbb{F}$ and they may represent highly complex shapes in the input space $\mathbb{R}^d$:

$$\{x : R^2 = k(x, x) - 2 \sum_{i=1}^{n} \alpha_i k(x, x_i) + \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j)\} \ , \tag{9}$$

depending on one's choice of the kernel function k. Kernels that have proven to be effective for data description include the Gaussian kernel $k(x_1, x_2) = \exp(-\|x_1 - x_2\|^2/\sigma^2)$ and the polynomial kernel $k(x_1, x_2) = (1 + \langle x_1, x_2 \rangle)^p$.

3 Pattern Classification via Single Spheres

In the above section, we have described how to construct a minimum bounding sphere to provide a compact description of a set of data, which are assumed to belong to the same class. For each class, such a sphere can be constructed without considering training data from other classes. In this section, we explore the possibility of using single spheres for pattern separation.

Given a set of training data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, instead of trying to find a sphere that provides a compact description of one class, for classification purposes, we want to find a sphere that encloses all examples from one class but excludes all examples from the other class, e.g., a sphere S with center c and radius R that encloses all positive examples and excludes all negative examples. In addition, we assume that sphere S separates the two classes with margin $2d$, i.e., it satisfies the following constraints:

$$R^2 - \langle x_i - c, x_i - c \rangle \geq d^2, \forall i \text{ such that } y_i = 1, \tag{10}$$

and

$$\langle x_i - c, x_i - c \rangle - R^2 \geq d^2, \forall i \text{ such that } y_i = -1, \tag{11}$$

where d is the shortest distance from the sphere to the closest positive and negative examples (see Fig. 1).

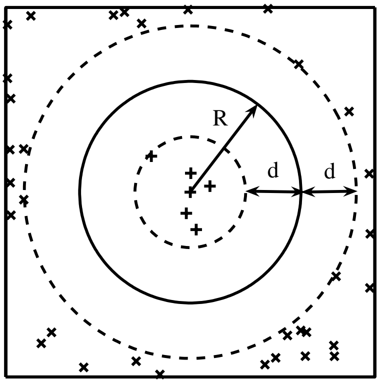


Fig. 1. Spherical classifier that maximizes the separation ratio

There may exist many spheres that satisfy the above constraints. Among many such spheres, it is natural that we seek to find a sphere that separates the training data with the maximum separation ratio, i.e.,

$$\max_{c,R,d} \frac{R+d}{R-d} \tag{12}$$

subject to

$$y_i(R^2 - \langle x_i - c, x_i - c\rangle) \geq d^2 \quad \forall i = 1, \ldots, n \ . \tag{13}$$

It is easy to show that maximization of the separation ratio $(R+d)/(R-d)$ is equivalent to minimization of R^2/d^2. The objective function R^2/d^2 is a nonlinear function of R^2 and d^2 and is hard to deal with directly. However, at any given point (R_0, d_0), R^2/d^2 can be approximated as:

$$\frac{R^2}{d^2} \approx \frac{R_0^2}{d_0^2} + \frac{1}{d_0^2}(R^2 - \frac{R_0^2}{d_0^2}d^2) \ . \tag{14}$$

Therefore, the problem of finding the sphere with maximum separation ratio can be reformulated as:

$$\min_{c,R,d} \ R^2 - Kd^2 \tag{15}$$

subject to

$$y_i(R^2 - \langle x_i - c, x_i - c\rangle) \geq d^2 \quad \forall i = 1, \ldots, n \ , \tag{16}$$

where $K = R_0^2/d_0^2 \geq 1$ is a constant that controls the ratio of the radius to the separation margin.

Introducing Lagrange multipliers $\alpha_i \geq 0$, one for each of the constraints in (16), we obtain the Lagrangian:

$$L = R^2 - Kd^2 - \sum_{i=1}^{n} \alpha_i [y_i(R^2 - \langle x_i - c, x_i - c\rangle) - d^2] \ . \tag{17}$$

The task is to minimize the Lagrangian L with respect to R, d, and c, and to maximize it with respect to α_i. Setting the partial derivatives to zero, we obtain

$$c = \sum_{i=1}^{n} \alpha_i y_i x_i \ , \tag{18}$$

which gives the center c of the sphere as a linear combination of training data x_i, and

$$\sum_{i=1}^{n} \alpha_i = K \tag{19}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 1 \ . \tag{20}$$

Substituting the new constraints into the Lagrangian (17), we obtain the following dual form of the quadratic programming problem:

$$\min_{\alpha_i, i=1,\ldots,n} \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \frac{1}{2} \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x_i \rangle \tag{21}$$

subject to

$$\alpha_i \geq 0, \quad \forall i = 1, \ldots, n \tag{22}$$

$$\sum_{i=1}^{n} \alpha_i = K \tag{23}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 1 \ . \tag{24}$$

It should be emphasized that, unlike the quadratic programming problems in Sect. 2 or in standard SVMs, the primal constrained optimization problem defined by (15) and (16) is non-convex. In fact, it is easy to see that the set of constraints (16) for all i such that $y_i = -1$ is non-convex. However, fortunately, the Lagrangian (17) is convex at the solution of the dual problem. Therefore, strong duality still holds and the solution of the dual problem provides an optimal solution of the primal problem.

Solving the dual problem, one obtains the coefficients $\alpha_i, i = 1, \ldots, n$. The center c of the optimal sphere can be obtained by Eq. (18). Similarly, the radius R can be determined from the KKT conditions by letting

$$R^2 = \frac{\min_{y_i=-1} \langle x_i - c, x_i - c \rangle + \max_{y_i=1} \langle x_i - c, x_i - c \rangle}{2} \ , \tag{25}$$

which leads to the following spherical decision function:

$$f(x) = \operatorname{sgn}\left(R^2 - (\langle x, x \rangle - 2 \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle + \sum_{i,j} \alpha_i \alpha_j \langle x_i, x_j \rangle) \right) \ . \tag{26}$$

In general, the solution to the above optimization problem may not exist because there is no such sphere in the input space that separates all the positive samples from the negative samples. Similarly to the SVDD case, we can apply the kernel trick here by replacing the inner products with suitable kernel functions. In effect, the maximum separation sphere is constructed in the feature space induced by the kernel. So far, we have only considered the case in which the data is separable by a sphere in the input space or in the feature space that is induced by the kernel. However, such a sphere may not exist, even in the kernel feature space. To allow for some classification errors, we introduce slack-variables $\xi_i \geq 0$ for $i = 1, \ldots, n$ to relax the constraints (13) with

$$y_i(R^2 - \langle x - c, x - c\rangle) \geq d^2 - \xi_i \ , \tag{27}$$

and consequently minimize the following objective function:

$$\min_{c,R,d,\xi_i,i=1,\ldots,n} R^2 - Kd^2 + C\sum_{i=1}^{n} \xi_i \ , \tag{28}$$

where the regularization constant C determines the trade-off between the empirical error and spherical separation margin term. Using the Lagrange multiplier method, we obtain the following dual problem in the kernel form:

$$\min_{\alpha_i,i=1,\ldots,n} \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j k(x_i, x_j) - \frac{1}{2}\sum_{i=1}^{n} \alpha_i y_i k(x_i, x_i) \tag{29}$$

subject to

$$0 \leq \alpha_i \leq C, \quad \forall i = 1, \ldots, n \tag{30}$$

$$\sum_{i=1}^{n} \alpha_i = K \tag{31}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 1 \ . \tag{32}$$

The above dual optimization problem can be solved using standard quadratic programming solvers, such as CPLEX, LOQO, MINOS and Matlab QP routines. Similarly to the standard SVMs, one can also use the sequential minimal optimization (SMO) method or other decomposition methods to speed up the training process by exploiting the sparsity of the solution and the KKT conditions [13,14,15].

It should be noted that separating data using spheres is a special case of separating data via ellipsoids, which results in a convex semi-definite program (SDP) that can be efficiently solved by interior point methods [16]. However, a drawback of the ellipsoid separation approach is that it cannot be easily extended by the kernel method, because the SDP problem cannot be expressed purely in inner products between input vectors. Therefore, both the decision boundaries it can generate and the problems it can solve are limited, unless special preprocessing is carried out prior to applying the ellipsoid separation method. On the other hand, using spheres combined with suitable kernels can produce more flexible decision boundaries than ellipsoids. Furthermore, SDP is limited in terms of the number of input dimensions it can effectively deal with.

4 Results and Discussion

We applied the method to both artificial and real-world data. The training algorithm was implemented based on the SMO method. Figure 2 displays a 2-D toy example and shows how different values of the parameter K lead to different solutions. The training examples of two classes are denoted as $+$'s and $\times$'s respectively in the figure. Clearly, there exist many spheres that can separate the training data in the 2-D input space. Therefore, for this dataset, no kernel trick was used, and the separating spheres were constructed directly in the input space using the standard definition of the Euclidean inner product. The three remaining plots show the results with three different values of the constant K. In each plot, three spheres (or their portions) are displayed. The darkest line represents the sphere with radius $R - d$. The lightest line represents the sphere with radius $R + d$. The line in between represents the separating sphere with radius R. The support vectors (the training examples with nonzero α values) are marked with small circles.

As we can see, increasing the value of K from 1 to 100, the shape of the decision surface changes from a sphere to a plane. When K is set to a small value, the algorithm finds a sphere that gives a compact description of the positive examples. For instance, when $K = 1$, the inner sphere (the sphere with radius $R - d$) coincides with the smallest sphere found by the SVDD method that encloses all the positive examples [12,17]. When K is set to a larger value, a larger sphere is found to contain the positive examples and the decision surface is more like a plane. Therefore, by adjusting the constant K that controls the ratio of the radius of the sphere to the separation margin, one can obtain a series of solutions from sphere-like decision boundaries to linear decision boundaries, including the solution of the SVDD method for data description and the solution of SVMs for classification.

Figure 3 shows the results of the spherical classifiers with a Gaussian kernel on another artificial dataset. The training data is generated randomly in a rectangular region. Training examples of the two classes, separated by $f_2 = \sin(\pi f_1)$, are denoted as $+$'s and $\times$'s respectively (see figure 3, upper-left plot). Clearly, there is no single sphere in the 2-D input space that can separate the two classes. We used a Gaussian kernel to map the data into a high dimensional feature space, in which the separating spheres were constructed. The remaining three plots show the results of the spherical classifier at different values of K. For better visualization, only training examples that correspond to the support vectors are shown in the three plots. The results demonstrate that a separating sphere was found in the feature space by adjusting the value of the constant K.

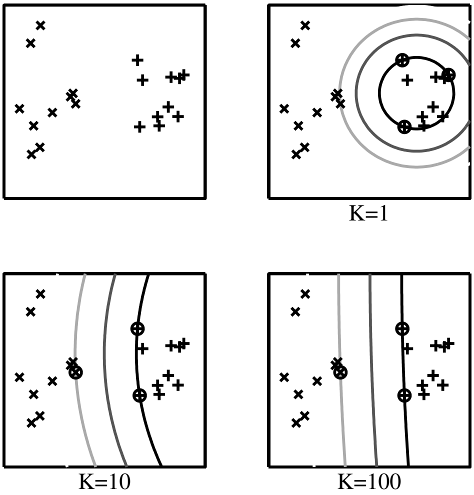


Fig. 2. Results of the spherical classifier on an artificial dataset at different values of K

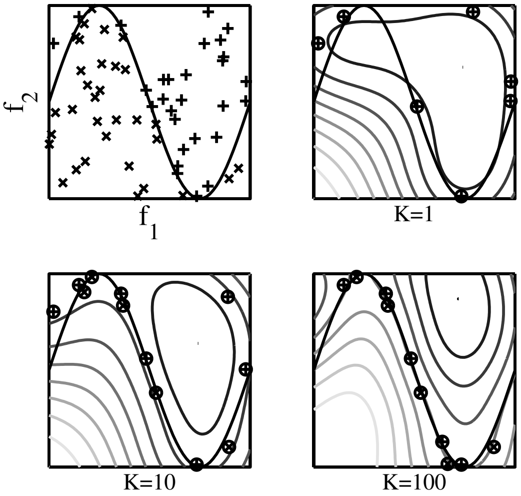


Fig. 3. Results of the spherical classifier (using a Gaussian kernel) on an artificial dataset. Top left: The training data and desired decision boundary; The rest: spheres of different radii mapped back onto the 2-D input space for three different values of K. The darker the line, the smaller the radius. The small circles around training examples indicate the support vectors.

We also tested the new algorithm and compared it to standard SVMs using several real-world datasets from the UCI machine learning repository [18]. For all the datasets, we used the 5-fold cross-validation method to estimate the generalization error of the classifiers. In the 5-fold cross-validation process, we ensured that each training set and each testing set were the same for both algorithms, and the same Gaussian kernel was used. The datasets used and the results obtained by the two algorithms are summarized in Table 1. The results of the spherical classifier and the SVM classifier both depend on the values of the kernel parameter σ and the regularization parameter C. In addition, the performance of the spherical classifier also depends on the value of K. In our tests, we set C to infinity for both algorithms, i.e., we only considered hard-margin spherical and hyperplane classifiers. On each dataset, the value of the kernel parameter σ was optimized to provide the best error rate of the SVM classifier, and the same value was used for the spherical classifier. As we can see, the spherical classifier achieves the same or slightly better results than SVMs on all 5 datasets.

Table 1. Comparison of Error Rates

Dataset	Sphere	SVM
Breast Cancer	4.26 (±1.73)	4.26 (±1.73)
Ionosphere	5.71 (±2.80)	6.00 (±2.86)
Liver	35.36 (±1.93)	36.23 (±5.39)
Pima	34.90 (±2.13)	35.03 (±2.20)
Sonar	10.73 (±1.91)	11.22 (±2.44)

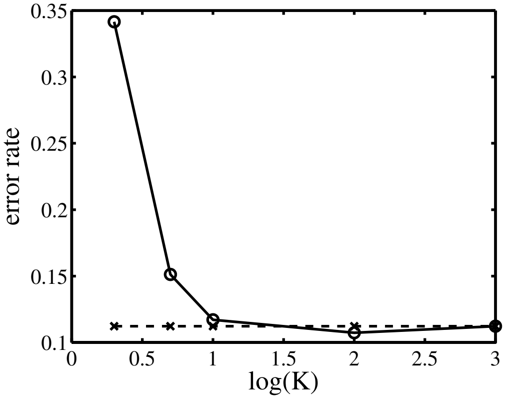


Fig. 4. Error rates of the spherical classifier on the sonar dataset for different values of K. The solid line represents the error rate of the spherical classifier. The dashed line is the error rate of the SVM classifier.

In Figure 4, we show a detailed comparison of the spherical classifier and the SVM classifier on the Sonar dataset. The solid line displays the error rates of the spherical classifier at different values of K. The dashed line gives the corresponding error rates of the support vector machine. Once again, the same kernel parameter σ was used for both algorithms, and the regularization parameter C was set to infinity. As we can see, the error rates of the spherical classifier decrease as the value of K increases. If K is set to be large enough, the result of the spherical classifier reaches that of the support vector machine, which is consistent with what we have observed in our toy examples.

From Table 1 and Fig. 4, we see that the spherical classifier yields comparable results as the support vector machine, demonstrating that it is suitable for real-world classification problems.

5 Conclusion

In this paper we explored the possibility of using single spheres for pattern classification. Inspired by the support vector machines and the support vector data description method, we presented an algorithm that constructs single spheres in the kernel feature space that separate data from different classes with the maximum separation ratio. By incorporating the class information of the training data, our approach provides a natural extension to the SVDD method of Tax and Duin, which computes minimal bounding spheres for data description (also called One-class classification).

By adopting the kernel trick, the new algorithm effectively constructs spherical boundaries in the feature space induced by the kernel. As a consequence, the resulting classifier can separate patterns that would otherwise be inseparable when using a single sphere in the input space. Furthermore, by adjusting the ratio of the radius of the separating sphere to the separation margin, a series of solutions ranging from spherical to linear decision boundaries can be obtained. Specifically, when the ratio is set to be small, a sphere is constructed that gives a compact description of the positive examples, coinciding with the result of the SVDD method; when the ratio is set to be large, the solution effectively coincides with the maximum margin hyperplane solution. Therefore, our method effectively encompasses both the support vector machines for classification and the SVDD method for data description. This feature of the proposed algorithm may also be useful for dealing with the class-imbalance problem. We tested the new algorithm and compared it to the support vector machines using both artificial and real-world datasets. The experimental results show that the new algorithm offers comparable performance on all the datasets tested. Therefore, our algorithm provides an alternative to the maximum margin hyperplane classifier.

References

1. Cooper, P.W.: The hypersphere in pattern recognition. Information and Control **5** (1962) 324–346
2. Cooper, P.W.: Note on adaptive hypersphere decision boundary. IEEE Transactions on Electronic Computers (1966) 948–949

3. Batchelor, B.G., Wilkins, B.R.: Adaptive discriminant functions. Pattern Recognition, IEEE Conf. Publ. **42** (1968) 168–178
4. Batchelor, B.G.: Practical Approach to Pattern Classification. Plenum, New York (1974)
5. Reilly, D.L., Cooper, L.N., Elbaum, C.: A neural model for category learning. Biological Cybernetics **45** (1982) 35–41
6. Scofield, C.L., Reilly, D.L., Elbaum, C., Cooper, L.N.: Pattern class degeneracy in an unrestricted storage density memory. In Anderson, D.Z., ed.: Neural Information Processing Systems. American Institute of Physics, Denver, CO (1987) 674–682
7. Marchand, M., Shawe-Taylor, J.: The set covering machine. Journal of Machine Learning Research **3** (2002) 723–746
8. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In Haussler, D., ed.: Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory. (1992) 144–152
9. Cortes, C., Vapnik, V.N.: Support vector networks. Machine Learning **20** (1995) 273–297
10. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York, NY (1998)
11. Schölkopf, B., Burges, C., Vapnik, V.: Extracting support data for a given task. In: Proceedings of First International Conference on Knowledge Discovery and Data Mining. (1995) 252–257
12. Tax, D.M.J., Duin, R.P.W.: Data domain description by support vectors. In Verleysen, M., ed.: Proceedings ESANN, Brussels, D. Facto Press (1999) 251–256
13. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B., Burges, C.J.C., Smola, A.J., eds.: Advances in Kernel Methods - Support Vector Learning, Cambridge, MA, MIT Press (1999) 185–208
14. Vapnik, V.N.: Estimation of Dependence Based on Empirical Data. Springer-Verlag, Berlin (1982)
15. Osuna, E., Freund, R., Girosi, R.: Support vector machines: training and applications, A.I. Memo AIM - 1602. MIT A.I. Lab (1996)
16. Glineur, F.: Pattern separation via ellipsoids and conic programming, Mémoire de D.E.A., Faculté Polytechnique de Mons, Mons, Belgium (Sept. 1998)
17. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J.: Estimating the support of a high-dimensional distribution. Neural Computation **13** (2001) 1443–1471
18. Blake, C., Merz, C.: UCI repository of machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html (1998)

SCALETRACK: A System to Discover Dynamic Law Equations Containing Hidden States and Chaos

Takashi Washio, Fuminori Adachi, and Hiroshi Motoda

I.S.I.R., Osaka University, 8-1, Mihogaoka, Ibaraki City, Osaka, 567-0047, Japan
washio@ar.sanken.osaka-u.ac.jp

Abstract. This paper proposes a novel system to discover simultaneous time differential law equations reflecting first principles underlying objective processes. The system has the power to discover equations containing hidden state variables and/or representing chaotic dynamics without using any detailed domain knowledge. These tasks have not been addressed in any mathematical and engineering domains in spite of their essential importance. Its promising performance is demonstrated through applications to both mathematical and engineering examples.

1 Introduction

A set of well known pioneering approaches of scientific law equation discovery is called BACON family [1]. They try to figure out a static equation on multiple quantities over a wide state range under a given laboratory experiment where quantities are actively controlled. Their drawback is the low likelihood to discover the law equations, since they do not use certain essential criteria to capture relations induced by the first principles. A law equation reflecting the first principle here is an observable, reproducible and concise relation satisfying generality, soundness and mathematical admissibility. The generality is to be widely observed in the objective domain of the equation, the soundness not to conflict with any observations and the mathematical admissibility to follow some constraints deduced from the invariance of the relation under various times, places and measurement expressions. Especially, the mathematical admissibility can be used to narrow down the equation formulae for the search. Some systems introduced unit dimension constraints and "*scale-type constraints*" to limit the search space to mathematically admissible equations [2,3,4]. Especially, the scale-type constraints have wide applicability since they do not need unit information of quantities. LAGRANGE addressed the discovery of "*simultaneous time differential law equations*" reflecting the dynamics of objective processes under "*passive observations*" where none of quantities are experimentally controllable [5]. Its extended version called LAGRAMGE introduced domain knowledge of the objective process to limit the search space within plausible law equations [6]. IPM having similar functions with LAGRAMGE further identified plausible law equations containing "*hidden state variables*" when the variables are known in

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 253–266, 2005.

the detailed domain knowledge [7]. PRET identified "*chaotic dynamics*" under similar conditions where very rich domain knowledge is available [8].

However, scientists and engineers can develop good models of the objective dynamics without using the discovery systems in many practical cases when detailed domain knowledge is available. Accordingly, the main applications of the discovery systems are to identify simultaneous time differential equations reflecting the first principles under passive observation and "*little domain knowledge*." One of such important applications is the discovery of "*hidden state variables*." In many problems, some state variables are not directly observed, and even the number of unobserved state variables is not known. Another important issue is the analysis of the observed data representing "*chaotic dynamics*." If the detailed domain knowledge on the dynamics underlying the chaos is given, some of the aforementioned systems can construct the dynamic equations appropriately representing the chaos. However, scientists can hardly grasp the dynamic laws on many chaotic behaviors based on their domain knowledge, since the background mechanisms of the chaos are usually very complex [9].

In this paper, we propose a novel scientific equation discovery system called SCALETRACK (SCALE-types and state TRACKing based discovery system) to discover a model of an objective process under the following requirements.

(1) The model is simultaneous time differential equations representing the dynamics of an objective process.
(2) The model is not an approximation but a plausible candidate to represent the underlying first principles.
(3) The model is discovered from passively observed data without using domain knowledge specific to the objective process.
(4) The model can include hidden state variables.
(5) The model can represent chaotic dynamics.

2 Outline

2.1 Basic Problem Setting

We adopt the following "*state space model*" of objective dynamics and measurement without loss of generality.

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{v}(t) \quad (\boldsymbol{v}(t) \sim N(0, \boldsymbol{\Sigma}_v)), \text{ and} \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{w}(t) \quad (\boldsymbol{w}(t) \sim N(0, \boldsymbol{\Sigma}_w)), \tag{2}$$

where the first equation is called a "*state equation*" and the second a "*measurement equation*." $\boldsymbol{x}$ is called a state vector, $\boldsymbol{f(x)}$ a system function, $\boldsymbol{v}$ a process noise vector, $\boldsymbol{y}$ a measurement vector, $\boldsymbol{C}$ a measurement matrix, $\boldsymbol{w}$ a measurement noise and t a time index. $\boldsymbol{f(x)}$, a model of the objective dynamics over its wide state range, is not limited to linear formulae in general, and any state transition of $\boldsymbol{x}$ can be represented by this formulation. $\boldsymbol{C}$, the model of measurement, is represented by a linear transformation matrix, because the measurement facilities are artificial and linear in most cases, and some state variables in $\boldsymbol{x}$ are often observed indirectly as their linear combinations through measurement variables in $\boldsymbol{y}$. If $\boldsymbol{C}$ is column full rank, the values of all state variables with the measurement noise are estimated by solving the measurement equation with $\boldsymbol{x}$. Otherwise, some state variables are not estimated within the measurement equation, and these variables are called "*hidden state variables*."

In the scientific law equation discovery, $\boldsymbol{f}(\boldsymbol{x})$ is initially unknown, and even $\boldsymbol{x}$ is not known correctly. Only a state subvector $\boldsymbol{x}'(\subseteq \boldsymbol{x})$ and a submatrix $\boldsymbol{C}'(\subseteq \boldsymbol{C})$ representing an artificial measurement facility are initially known to relate $\boldsymbol{x}'$ with $\boldsymbol{y}$ as $\boldsymbol{y} = \boldsymbol{C}'\boldsymbol{x}'$. To derive $\boldsymbol{C}$ from $\boldsymbol{C}'$, the number of missing state variables, *i.e.*, the difference between the dimensions of $\boldsymbol{x}$ and $\boldsymbol{x}'$, must be estimated. Thus, SCALETRACK identifies the number of elements in $\boldsymbol{x}$ including hidden state variables based on passively observed data at first. Then, it searches plausible candidates of $\boldsymbol{f}(\boldsymbol{x})$ reflecting the first principles from the data.

2.2 Entire Approach

The entire approach of SCALETRACK is outlined in Figure 1. Given a set of measurement data, the dimension of $\boldsymbol{x}$ is identified through a statistical analysis called "*correlation dimension analysis*." Once the dimension is known, all possible combinations of scale-types of the elements in $\boldsymbol{x}$ are enumerated based on scale-type constraints, the known measurement submatrix $\boldsymbol{C}'$ and the known scale-types of the elements in $\boldsymbol{y}$. Then, for every combination, the candidate formulae of a state equation admissible to the scale-type constraints are generated. Subsequently, through a set of state tracking simulations called "SIS/RMC filter" combined with parameter search on the given measurement data, the parameter values in every candidate formula are estimated. Finally, some candidates providing highly accurate tracking in terms of "*Mean Square Error* (MSE)" are selected as the discovered dynamic models of the objective process. The details of each step in Figure 1 are described in the following section.

3 Methods

3.1 Estimating Dimension of $\boldsymbol{x}$

"*Correlation dimension analysis*" estimates the dimension of $\boldsymbol{x}$, $dim(\boldsymbol{x})$, from given measurement data $\boldsymbol{y}$ over n sampling time steps [9]. Given an element y_h $(h = 1,..,dim(\boldsymbol{y}))$ of $\boldsymbol{y}$, let τ_h be the minimum time step lag that the time lagged autocorrelation of $y_h(t)$ becomes 0 as follows.

$$\tau_h = argmin_{\tau \in [1,n]}\{\frac{1}{n}\sum_{t=1}^{n-\tau}(y_h(t) - \bar{y}_h)(y_h(t+\tau) - \bar{y}_h) \simeq 0\}, \tag{3}$$

where $\bar{y}_h$ is the time average of $y_h(t)$ over $[1,n]$. τ_h is the time steps within that the local dependency among the observed states is vanished. Then the following time lagged vectors of length m are constructed from y_h.

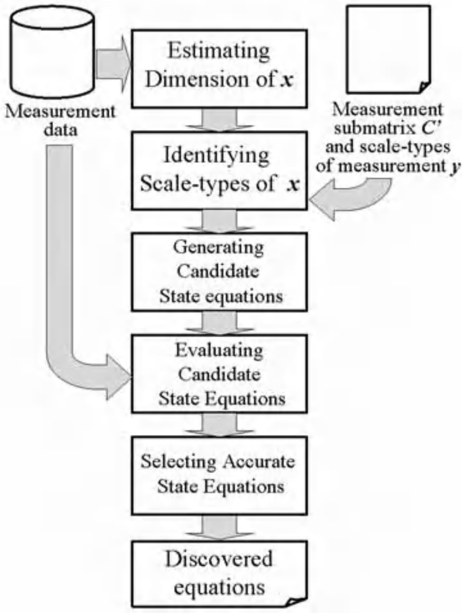

Measurement data
Estimating Dimension of x
Identifying Scale-types of x
Measurement submatrix C' and scale-types of measurement y
Generating Candidate State equations
Evaluating Candidate State Equations
Selecting Accurate State Equations
Discovered equations

Fig. 1. Outline of SCALETRACK

$$Y_h^m(1) = [y_h(1), y_h(1+\tau_h), \ldots, y_h(1+(m-1)\tau_h)]$$
$$\cdots\cdots\cdots$$
$$Y_h^m(n-(m-1)\tau_h) = [y_h(n-(m-1)\tau_h), y_h(n-(m-2)\tau_h), \ldots, y_h(n)]$$

If m is sufficiently large, each of these vectors reflects a global relation among the states, since the time intervals among the elements in a vector are equal to or longer than τ_h. Then the following correlation integral in the time lagged phase space is calculated.

$$R_h^m(r) = \frac{2}{n'(n'-1)}[\text{number of } (i,j)\text{s; } \mathit{\Delta} Y_h^m(i,j) < r], \tag{4}$$

where $n' = n-(m-1)$, $1 \leq i,j \leq n'$ and $\mathit{\Delta} Y_h^m(i,j) = |Y_h^m(i) - Y_h^m(j)|$. $R_h^m(r)$ represents the density of states in the space, and shows the following power law relation in general over the range of r covering the state distribution.

$$R_h^m(r) \propto r^{\nu_h(m)}, \tag{5}$$

where $\nu_h(m)$ is called a "*correlation exponent.*" Theoretically it is an approximation of the fractal dimension of the global state distribution which is equivalent to $dim(\boldsymbol{x})$ under the condition of $m \geq 2dim(\boldsymbol{x})+1$. $dim(\boldsymbol{x})$ is estimated through the least square fitting of Eq.(5) to $R_h^m(r)$s derived by Eq.(4) under a sufficiently large m. $\nu_h(m)$ is computed for each y_h $(h = 1, .., dim(\boldsymbol{y}))$, and the nearest integer of its maximum, $\nu_{max}(m)$, among them is used for $dim(\boldsymbol{x})$, since some measurement variables may miss the behaviors of some state variables.

3.2 Identifying Scale-Types of $\boldsymbol{x}$

Once $dim(\boldsymbol{x})$ is known, the "*scale-type*" of each element of $\boldsymbol{x}$ is identified for the candidate $f(\boldsymbol{x})$ generation in the next step. This is done based on "*scale-type constraints*" [10] and the scale-types of elements of $\boldsymbol{y}$. Representatives of quantitative scale-types are ratio scale and interval scale. Examples of the ratio scale quantities are physical mass and absolute temperature where each has an absolute origin. The admissible unit conversion of the ratio scale follows $x' = \alpha x$. Examples of the interval scale quantities are temperature in Celsius and sound pitch where the origins of their scales are not absolute and arbitrary changed by human's definitions. The admissible unit conversion of the interval scale follows $x' = \alpha x + \beta$. Though the scale-type is strongly related with the unit dimension, they are different each other.

As noted in the previous section, only a state subvector $\boldsymbol{x'}(\subseteq \boldsymbol{x})$ is measured by $\boldsymbol{y}$ through a measurement facility $\boldsymbol{C'}(\subseteq \boldsymbol{C})$ as $\boldsymbol{y} = \boldsymbol{C'x'}$. Because the structure of the facility is independent of the units of the elements of $\boldsymbol{x'}$ and $\boldsymbol{y}$, $\boldsymbol{C'}$ is invariant against the change of their units. Then the following theorem holds.

Linear Formula Theorem. *Let $\boldsymbol{x'}$ be a known state subvector of $\boldsymbol{x}$, y_h an element of a measurement vector $\boldsymbol{y}$ and $\boldsymbol{x'_h}$ a state subvector of $\boldsymbol{x'}$ where each $x_i \in \boldsymbol{x'_h}$ has a nonzero (h,i)-element, c_{hi}, in the known measurement submatrix $\boldsymbol{C'}$. The scale-types of x_is in $\boldsymbol{x'_h}$ are constrained by the scale-type of y_h and the following rules.*

(1) If y_h is a ratio scale, all x_is are ratio scales, or more than one x_i are interval scales and the rest ratio scales.

(2) If y_h is an interval scale, one x_i at least is an interval scale and the rest ratio scales.

Proof. Because of the relation $\boldsymbol{y} = \boldsymbol{C'x'}$, $y_h = \sum_{x_i \in \boldsymbol{x'_h}} c_{hi}x_i$ holds. Let the set of interval scale quantities in $\boldsymbol{x'_h}$ be I_h. Every $x_i \in I_h$ follows the admissible unit conversion $x'_i = \alpha_i x_i + \beta_i$, and every x_i in the rest, *i.e.*, ratio scales, follows $x'_i = \alpha_i x_i$. When y_h is a ratio scale, it follows $y'_h = \alpha y_h$. Because of the invariance of $\boldsymbol{C'}$, $y'_h = \sum_{x'_i \in \boldsymbol{x'_h}} c_{hi}x'_i$ holds. By substituting the admissible unit conversions and $y_h = \sum_{x_i \in \boldsymbol{x'_h}} c_{hi}x_i$ to this linear relation, the following is obtained.

$$\sum_{x_i \in \boldsymbol{x'_h}} \alpha c_{hi} x_i = \sum_{x_i \in \boldsymbol{x'_h}} c_{hi}\alpha_i x_i + \sum_{x_i \in I_h} c_{hi}\beta_i$$

Because this is an identity equation for every $x_i \in \boldsymbol{x'_h}$, $\alpha_i = \alpha$ for every x_i and $\sum_{x_i \in I_h} c_{hi}\beta_i = 0$ hold. If I_h is empty, the last relation trivially holds. If I_h has only a unique x_i, $\beta_i = 0$ must hold, and this is contradictory to the interval scale $x_i \in I_h$. If I_h has more than one x_i, the last relation can hold for non-zero β_is while β_is are mutually dependent in the relation. This concludes the rule (1). When y_h is an interval scale, it follows $y'_h = \alpha y_h + \beta$. Through the similar discussion with the rule (1), $\sum_{x_i \in I_h} c_{hi}\beta_i = \beta$ hold. If I_h is empty, $\beta = 0$ must hold, and this is contradictory to the interval scale y_h. If I_h is not empty, this relation can hold for non-zero β_is and β while they are mutually dependent in the relation. This concludes the rule (2). ∎

Based on this theorem and the scale-types of all $y_h \in \boldsymbol{y}$, a set of constraints on the scale-types of all $x_i \in \boldsymbol{x'}$ is obtained. Because the scale-types of all $x_i \in \boldsymbol{x}$ which are not in $\boldsymbol{x'}$ are unknown, they can be either ratio or interval scale. Then, every admissible combination (R_x, I_x) where R_x is a set of ratio scale state variables and I_x a set of interval scale state variables in $\boldsymbol{x}$ satisfying these constraints are enumerated by using a simple search. Though this search is combinatorial, it is tractable in practice as far as the dimension of $\boldsymbol{x}$ is not very large.

3.3 Generating Candidate State Equations

"*Extended Product Theorem*" [4] provides a basis of the candidate generation of state equations. This theorem comes from the invariance of the formula shape against the unit conversions and the scale-type constraints similarly to the aforementioned Linear Formula Theorem, and it has been used in several law equation discovery systems in the past. The following is the theorem where some notions are adapted to our descriptions.

Extended Product Theorem. *Given a combination (R_x, I_x) for $\boldsymbol{x}$, the state variables have the following relation.*

$$\dot{x}_i = \prod_{x_j \in R_x} |x_j|^{\alpha_j} \prod_{I_k \subseteq (I_x - I_g)} (\sum_{x_j \in I_k} \beta_{kj}|x_j| + \beta_k)^{\alpha_k} \prod_{x_j \in I_g} \exp(\beta_{gj}|x_j|)$$

where $x_i \in R_x \cup I_x$, all coefficients are constants, I_g a subset of I_x, and $\{I_k\}$ a covering of $(I_x - I_g)$.

In a state equation $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t))$, all elements in $\dot{\boldsymbol{x}}$ are ratio scales, since the time derivative of an element of $\boldsymbol{x}$ is the difference of two states divided by a time interval in essence. The formulae following this theorem are called "*regime*s" having the invariance against the unit conversions. Since this is a required character of the formulae to represent the first principles, the candidates have high plausibility to be law equations. Under ratio scale time derivatives $\dot{\boldsymbol{x}}$ and a given combination (R_x, I_x), the multiple candidates of a state equation are enumerated based on this theorem. The set of combinations of (R_x, I_x) derived in the previous step provides a set of many candidate state equations, CSE.

3.4 Evaluating Candidate State Equations

Once CSE for an objective process is provided in the previous step, a fitting error $E(c)$ of every candidate $c \in CSE$ under given measurement data is evaluated through adjustment of its coefficients and state tracking.

Searching for Power Coefficients. As shown in Extended Product Theorem, the formulae of the state equations have two types of constants, *i.e.*, power coefficients αs and proportional coefficients βs. The search space of a power

Table 1. Nonlinearity of T^{α}.

case of α		range of T	nonlinearlity
range	parity		
$\alpha > 1$	α is even.	$T \geq 0$	monotonic increase
		$T < 0$	monotonic decrease
$0 < \alpha < 1$	$1/\alpha$ is even.	$T \geq 0$	monotonic increase
		$T < 0$	not admissible
$\alpha > 0$	α or $1/\alpha$ is odd.	$T \geq 0$	monotonic increase
		$T < 0$	monotonic increase
$\alpha = 0$			1
$\alpha < 0$	α or $1/\alpha$ is odd.	$T \geq 0$	monotonic decrease
		$T < 0$	monotonic decrease
$-1 < \alpha < 0$	$1/\alpha$ is even.	$T \geq 0$	monotonic decrease
		$T < 0$	not admissible
$\alpha < -1$	α is even.	$T \geq 0$	monotonic decrease
		$T < 0$	monotonic increase

where $T \to 0 \Rightarrow T^{\alpha} \to 0 (\alpha > 0)$ and $T^{\alpha} \to \pm\infty (\alpha < 0)$.

coefficient α is limited to small integers, within $[-5, 5]$ for instance, and their inverses. This is because the power coefficients reflect the dimensions of space and units where the objective process operates, and their complexities are limited. Moreover, given a term T having a power coefficient α in the formulae of Extended Product Theorem, the range and the parity of α strongly affect the nonlinearity of T^{α} as shown in Table 1. Because of these discrete characteristics of α, the standard approaches for continuous and nonlinear optimization such as gradient descent method are not applicable. Instead, for every combination of the cases over all αs appearing in the candidate c, a monotonic and discrete search on integer αs is applied to reduce the fitting error $E(c)$. Because the number of αs in c is not very large, this part does not cause severe combinatorial explosion.

Searching for Proportional Coefficients. The search of the proportional coefficients βs minimizing $E(c)$ under every combination of αs provided in abovementioned scheme is performed. We experienced in our preliminary study that the standard nonlinear optimization of βs such as gradient descent method again does not converge to their right values within tractable time, because the influence of some βs to $\dot{x}_i$ can be very small under some αs. Accordingly, the following Golden Ratio Search [11] which is a well-known opportunistic line search without using the quantitative gradient information has been applied to βs. Under a combination of values of all αs and a combination of default values of all βs appearing in c, given initial upper bound β_j^u and lower bound β_j^l of β_j in c, $E(c)$s are evaluated on the following β_j^1 and β_j^2 by the state tracking which will be described shortly.

$$\beta_j^1 = \beta_j^l + r(\beta_j^u - \beta_j^l), \text{ and} \tag{6}$$

$$\beta_j^2 = \beta_j^u - r(\beta_j^u - \beta_j^l), \tag{7}$$

where $r = (3 - \sqrt{5})/2$ is the golden ratio. Let $E(c)$s evaluated on β_j^1 and β_j^2 be $E(c|\beta_j^1)$ and $E(c|\beta_j^2)$ respectively. If $E(c|\beta_j^1) \geq E(c|\beta_j^2)$ then $\beta_j^1 \rightarrow \beta_j^l$, $\beta_j^2 \rightarrow \beta_j^1$ and calculate new β_j^2 by Eq.(7), else $\beta_j^2 \rightarrow \beta_j^u$, $\beta_j^1 \rightarrow \beta_j^2$ and calculate new β_j^1 by Eq.(6). This rule is applied iteratively until $|\beta_j^2 - \beta_j^1|$ becomes less than a threshold ϵ. After this convergence, the converged value becomes a new default value of β_j. Subsequently, another β in c is selected in place of β_j, and this Golden Ratio Search is repeated until the default values of all βs in c becomes stable. Finally, the estimated βs are rounded off to integers when the values are close enough to the integers within the statistically expected estimation errors, since the parameters tend to be integers in many physical processes. After obtaining values of all βs for each combination of values of all αs, the unique combination of values of all αs, A_c, and that of all βs, B_c, providing the minimum $E(c)$ is chosen to be the coefficients of c.

State Tracking. Given a time series of measurement vector $\boldsymbol{y}(t)$s, a candidate state equation c and its $A_c \cup B_c$, the fitting error $E(c)$ is evaluated through state tracking. The recent massive increase in computational power became to allow the introduction of direct and sequential Monte Carlo integration of the state probability distributions within Bayesian framework. This approach is called "*Sequential Importance Sampling/Resampling Monte Carlo filter* (SIS/RMC filter)" [12], and can track the states generated in c without introducing any essential approximation. This state tracking has many advantages comparing with the other nonlinear state tracking approaches such as the conventional Extended Kalman Filter [13] and the qualitative reasoning based PRET [8]. The former using the linearization of the state equations does not work well when the equations include some singular points and/or some state regions having strong sensitivity to the tracking error. The latter faces a combinatorial explosion of qualitative states when the dimension and/or the complexity of the state space structure are high. In contrast, SIS/RMC filter does not require any approximation to be spoiled by the singularity and the strong nonlinearity, and does not face the combinatorial explosion of the states to be considered.

Because of the space limit, readers should refer the literature [12] to learn the background theory of SIS/RMC filter. In this paper, only the procedure of the state tracking adapted to our basic problem setting is indicated. The SIS/RMC filter is represented by the following procedures where the probabilities $p(\boldsymbol{x}(t)|\boldsymbol{x}(t-1), \boldsymbol{y}(t))$ and $p(\boldsymbol{y}(t)|\boldsymbol{x}(t-1))$ are defined by $\boldsymbol{y}(t)$, c and its $A_c \cup B_c$.

1 Importance sampling

(1-1) For $i = 1, ..., N$, sample $\tilde{\boldsymbol{x}}^{(i)}(t) \sim p(\boldsymbol{x}(t)|\boldsymbol{x}^{(i)}(t-1), \boldsymbol{y}(t))$.

(1-2) For $i = 1, ..., N$, evaluate the importance weights:
$w^{*(i)}(t) = w^{*(i)}(t-1)p(\boldsymbol{y}(t)|\boldsymbol{x}^{(i)}(t-1))$.

(1-3) For $i = 1, ..., N$, normalize the importance weights: $\tilde{w}^{(i)}(t) = \frac{w^{*(i)}(t)}{\sum_{j=1}^{N} w^{*(j)}(t)}$.

(1-4) Let MAP estimation, $\tilde{\boldsymbol{x}}(t)$, be $\tilde{\boldsymbol{x}}^{(i)}(t)$ having the maximum $\tilde{w}^{(i)}(t)$.

(1-5) $N_{eff} = \frac{1}{\sum_{i=1}^{N}(\tilde{w}_k^{(i)})^2}$.

(1-6) If $N_{eff} \geq N_{thres}$ then $\boldsymbol{x}^{(i)}(t) = \tilde{\boldsymbol{x}}^{(i)}(t)$ for $i = 1, ..., N$, $t = t + 1$ and go to 1. Otherwise go to 2.

2 Resampling

(2-1) Generate random integers $j(i)$ $(i = 1, ..., N)$ in proportion to the probabilities $\tilde{w}^{(l)}(t)$ $(l = 1, ..., N)$ so that l having larger $\tilde{w}^{(l)}(t)$ appears more as $j(i)$.

(2-2) $x^{(i)}(t) = \tilde{x}^{j(i)}(t)$, $w^{(i)}(t) = 1/N$ for $i = 1, ..., N$, $t = t + 1$ and go to 1.

In the importance sampling, many $\tilde{\boldsymbol{x}}^{(i)}(t)$s called "*particles*" derive "*Maximum A Posteriori* (MAP)" estimation of the state vector in concert with the normalized weight $\tilde{w}^{(i)}(t)$. An index N_{eff} monitors the ratio of probable particles having high weights. When the ratio becomes lower than a predefined threshold N_{thres}, resampling is applied to increase the probable particles.

Once the MAP estimation $\tilde{\boldsymbol{x}}(t)$s are obtained over $t = 1, ..., n$ time steps, the time series of $\tilde{\boldsymbol{y}}(t)$s $(t = 1, ..., n)$ are estimated via Eq.(2). Then the fitting error $E(c)$ can be evaluated by the following "*Mean Square Error* (MSE)."

$$E(c) = \frac{1}{n} \sum_{t=1}^{n} |\boldsymbol{y}(t) - \tilde{\boldsymbol{y}}(t)|^2$$

3.5 Selecting Accurate State Equations

The previous steps provide CSE and $< c, A_c \cup B_c, E(c) >$ for all $c \in CSE$. The solutions $< c, A_c \cup B_c, E(c) >$ having the top K accuracy, *i.e.* the K least $E(c)$s, are selected as discovered dynamic state equations in large CSE. The value of K is empirically chosen according to the complexity of the objective process and the quality of measurement data. $K = 5$ is used throughout this paper to check the variation of the search space.

4 Result

4.1 Implementation

The evaluation of candidate state equations by the SIS/RMC filter is the most time consuming step. Any search can not be skipped, since the search space is discrete and nonmonotonic. We experienced that one run of a stand alone SCALETRACK to discover a simple state equation took more than weeks even if we used an efficient algorithm. Accordingly, the current SCALETRACK introduced a simple grid computing framework using a PC cluster consisting of a control server and 10 clients, where the server has an AthlonXP1900+ 1.6GHz CPU and 2GB RAM, and each client has an AthlonXP3000+ 2.7GHz and 512MB RAM. The server computes the first three steps and then allocates the task to evaluate 10% of candidate state equations to each computer. Because this task is mutually independent, and occupies the most of computation of SCALETRACK, this implementation accelerates the run speed almost 10 times.

4.2 Basic Performance Evaluation

Basic performance of SCALETRACK in terms of scale-types of state variables, hidden state variables and measurement noise levels is evaluated by using the following two artificial formulae of two dimensions.

$$\left.\begin{array}{l}\dot{x}_1(t) = x_1(t)x_2(t)\\ \dot{x}_2(t) = -0.5x_1(t)\end{array}\right\} RR,$$

where $y_1 = x_1$ and $y_2 = x_2$ are ratio scale.

$$\left.\begin{array}{l}\dot{x}_1(t) = 0.4x_1(t)(x_2(t) + 0.2)\\ \dot{x}_2(t) = -0.1(x_2(t) + 0.6)\end{array}\right\} RI,$$

where $y_1 = x_1$ is ratio scale and $y_2 = x_2$ interval scale. The measurement data were generated by the simulations under one time step $\Delta t = 0.005$ and total steps $n = 600$. Empirically, m in the correlation dimension analysis and N in the state tracking were chosen to be 7 and 500 respectively. The process noise is set to be 0 to check the pure effect of the measurement noise. These settings were used in every demonstration in the rest of this paper.

Table 2. Basic Performance

case	$\nu_{max}(7)$	ct (h)	$\sigma_w(\%)$				
			0.1	0.5	1.0	2.0	5.0
RR	2.21	1.5	+	±	±	±	−
RRH	2.21	5.5	±	±	−	−	−
RI	2.19	4.0	+	±	±	±	−
RIH	2.19	5.5	+	±	−	−	−

ct is a required comp. time and σ_w a measurement noise level.

Table 2 shows the result of the evaluation. The case names, RR and RI, in the table correspond to the above two state equations, and RRH and RIH are the cases where the second measurement variable y_2 is not available, and hence x_2 is hidden. The correlation dimension analysis properly estimated the dimension of state vectors as nearly 2 in each case, and thus two state variables were assumed in the subsequent steps. The computation times required for RRH, RI and RIH were longer than that of RR, because the variety of admissible formulae containing interval scale variables is larger than that of ratio scale variables. The result in that the formula having the correct shape is top ranked by the accuracy is marked by + in the table. If the formula having the correct shape is derived within the top five solutions, it is marked by ±, otherwise it is marked by −. The table shows that almost 2.0% relative noise is acceptable for the discovery of the correct formulae, if all state variables are measured. On the other hand, noise less than 1.0% is required to discover the correct formulae, if a hidden state variable exists. Similar results were obtained under the other n samplings more than a hundred. Since $0.5 - 2.0\%$ noise is widely seen in many scientific and engineering process measurements, the basic performance of SCALETRACK is considered to be acceptable for practical use, though further improvements on the noise robustness is needed in future study.

4.3 Discovery of Circuit Dynamics

SCALETRACK has been applied to synthetic data of an electric circuit consisting of LCs and a Field Effect Transister (FET) as shown in Figure 2 . Its state equation is represented as follows.

$$\dot{V}_I(t) = -\frac{I(t)}{C_1} = -100I(t),\ \dot{I}(t) = \frac{V_I(t)}{L} = 50V_I(t),\ \text{and}$$
$$\dot{V}_F(t) = \frac{V_I(t)V_F(t)}{rC_2} = 250V_I(t)V_F(t),$$

where the definitions of $V_I, I, V_F, L = 20mH, C_1 = 10mF$ and $C_2 = 1mF$ are clear in the figure, and $r = 4.0\Omega V$ is a voltage-resistance coefficient of the FET. All state variables are ratio scale, and can be measured via corresponding ratio scale measurement variables respectively. The measurement data were sampled under one time step $\Delta t = 0.001$, total time steps $n = 800$ and relative noise level $\sigma_w = 1.0\%$. Because $\nu_{max}(7) = 2.94$ was obtained in the correlation dimension analysis, the state equations consisting of three state variables were searched.

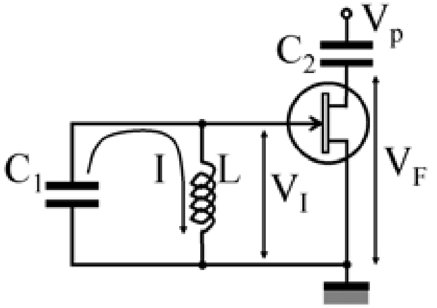


Fig. 2. An LC and FET Circuit

When every state variables are directly measured, the computation time was 18.5 hours, and the following equation having the best accuracy was derived.

$$\dot{V}_I(t) = -133.3I(t),\ \dot{I}(t) = 60.2V_I(t),\ \text{and}$$
$$\dot{V}_F(t) = 249.0V_I(t)V_F(t).$$

Though the values of coefficients are moderately different from the the originals, the entire shape of the formulae is identical. Next, the measurement of I was omitted to make I a hidden state variable. The computation time was 24 hours.

In this case, the following correct formula except the discrepancy of coefficient values showed up within the solutions having top five accuracies.

$$\dot{V}_I(t) = -26.9I(t),\ \dot{I}(t) = 298.0V_I(t),\ \text{and}$$
$$\dot{V}_F(t) = 250.0V_I(t)V_F(t).$$

These results indicate that SCALETRACK has ability to discover state equations of engineering objects having three dimensional dynamics.

4.4 Discovery of Chaos

The future state of a chaotic process will never be identical with its past state, and thus the state changes as if it is partially at random. Due to this nature, the state of the process gradually loses the dependency on its past state in a long term, and this makes harder to identify the dynamic equations governing the process. Nevertheless the trajectory of the state evolution is determined by the current state in the chaotic dynamics. Accordingly, the dynamic equation of the process can be discovered, if the state of the process is observed in sufficiently short sampling intervals comparing with the term length in which the state dependency dies out. Because the maximum term length of the state dependency is known by τ_h of Eq.(3) introduced in the aforementioned correlation analysis, the appropriate sampling interval can be easily known.

Under this consideration, the identification of chaotic dynamics was attempted. The state equation to be discovered is the following Altered Rossler Chaos.

$$\dot{x}_1 = -x_2 - x_3, \dot{x}_2 = x_1 + 0.36 * x_2,\ \text{and}$$
$$\dot{x}_3 = 0.01 * (x_1 - 4.5) * (x_1 + 1000 * x_3 - 4.5).$$

This has an attractor in a (x_1, x_2, x_3)-phase space as depicted in Figure 3. All state variables are interval scale, and can be measured through the corresponding interval scale measurement variables respectively. The measurement data

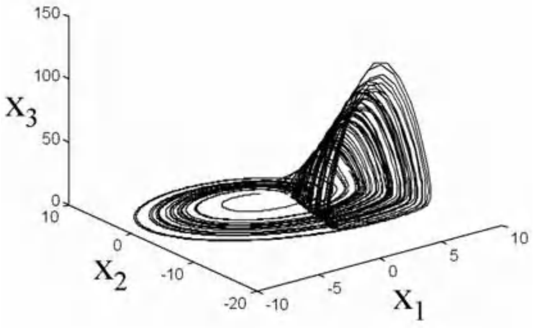


Fig. 3. An Attractor of Altered Rossler Chaos

were simulated under one time step $\Delta t = 0.001$, total time steps $n = 1500$ and relative noise level $\sigma_w = 1.0\%$. Because $\nu_{max}(7) = 3.33$ was obtained in the correlation dimension analysis, SCALETRACK searched for state equations consisting of three state variables. The computation time was 15.0 hours, and SCALETRACK resulted the following most accurate state equation. This formula has an identical shape with the original except some discrepancies of coefficients. This result indicates the high ability of SCALETRACK to discover the dynamic models of chaotic behaviors reflecting the underlying first principles.

$$\dot{x}_1 = -x_2 - x_3, \dot{x}_2 = x_1 + 0.33 * x_2, \text{ and}$$
$$\dot{x}_3 = 0.064 * (x_1 - 6.34) * (x_1 + 1002 * x_3 - 4.75).$$

5 Discussion

SCALETRACK is the first discovery system which introduced a state tracking approach in the search mechanism. In the demonstrations, the discrepancies of coefficient values are frequently observed. This may be because the particles and the weights are updated to follow the time series of $\boldsymbol{y}(t)$s at each time step in the SIS/RMC filter. This correction derives the robustness of the state tracking, but reduces the precision of the coefficient values. This may be a reason why the standard approaches for continuous and nonlinear optimization of the coefficients such as gradient descent do not perform well in the search. Another observation is that the candidate state equations top ranked by the accuracy often have formulae shapes different from the originals. This also may be due to the robustness of the state tacking against the modeling error in addition to the existence of many local minima of the accuracy in the nonlinear search space. Accordingly, the performance improvement of SCALETRACK is expected by introducing less robust state tracking, and this is a future research topic.

Another remaining issue is the current limitation of the search space. The search space of SCALETRACK is currently limited to a class of equation formulae called "*regime*"s specified by Extended Product Theorem. Although this class captures ample law equation formulae, another class of dynamic equations called "*ensemble*"s which are coupled with dimensionless variables are known not to be covered by this class. Further extension of criteria and algorithm for the search must be introduced in future while maintaining the tractability of the computation.

Introducing further valid constraints to narrow down the formulae within the law equations may enhance the plausibility of the discovered equations while reducing the search space. One of the candidate constraints is the relational templates representing conservation and flow of entities and interactions similar to Bond-Graph approach [14]. Though this type of constraints significantly contributes to the plausibility and the search space reduction in some domains including physics, they may not be applied to the wider domains such as economy and psychology where these templates do not hold, and thus the discovery systems become domain dependent. Introduction of new search constraints must be explored by carefully considering both the domain dependency and the efficiency.

6 Conclusion

SCALETRACK achieved three advantages which has not been addressed in any past work of mathematics, physics and engineering not limited to scientific discovery. The first is the discovery of first principle based simultaneous time differential equations without using detailed domain knowledge. The second is the discovery of hidden state variables. The third is the discovery of chaotic dynamics. These advantages are essentially important in many scientific and engineering fields due to the wide existence of such dynamics in nature.

References

1. Langley, P.W., Simon, H.A., Bradshaw, G.L., Zytkow, J.M.: Scientific Discovery; Computational Explorations of the Creative Process. MIT Press, Cambridge, Massachusetts (1987)
2. Koehn, B.W., Zytkow., J.M.: Experimeting and theorizing in theory formation. In: Proceedings of the International Symposium on Methodologies for Intelligent Systems, Knoxville, Tennessee, ACM SIGART Press. (1986) 296–307
3. Falkenhainer, B.C., Michalski, R.S.: Integrating quantitative and qualitative discovery: The abacus system. Machine Learning **1** (1986) 367–401
4. Washio, T., Motoda, H.: Discovering admissible models of complex systems based on scale-types and identity constraints. In: Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, Nagoya, Japan (1997) 810–817
5. Dzeroski, S., Todorovski, L.: Discovering dynamics: from inductive logic programming to machine discovery. Journal of Intelligent Information Systems **4** (1995) 89–108
6. Todorovski, L., Dzeroski, S.: Declarative bias in equation discovery. In: Proceedings of the Fourteenth International Conference on Machine Learning, San Mateo, California, Morgan Kaufmann (1997) 376–384
7. Langley, P., George, D., Bay, S., Saito, K.: Robust induction of process models from time-series data. In: Proceedings of the Twentieth International Conference on Machine Learning, Menlo Park, California, The AAAI Press (2003) 432–439
8. Bradley, E.A., O'Gallagher, A.A., Rogers, J.E.: Global solutions for nonlinear systems using qualitative reasoning. Annals of Mathematics and Artificial Intelligence **23** (1998) 211–228
9. Berge, P., Pomeau, Y., Vidal, C.: Order in Chaos - For understanding turbulent flow. Hermann, Paris, France (1984)
10. Luce, D.R.: On the possible psychological laws. Psychological Review **66** (1959) 81–95
11. Luenberger, D.G.: Linear and Nonlinear Programing. Ed. Adison-Wesley, Cambridge, Massachusetts (1989)
12. Doucet, A., Godsill, S., Andrieu, C.: On sequential monte carlo sampling methods for bayesian filtering. Statistics and Computing **10** (2000) 197–208
13. Haykin, S.S.: Kalman Filtering and Neural Networks. John Wiley & Sons, Inc., Hoboken, New Jersey (2001)
14. Gawthrop, P.J., Smith, L.S.: Metamodelling: Bond Graphs and Dynamic Systems. Prentice-Hall, Englewood Cliffs, New Jersey (1996)

Exploring Predicate-Argument Relations for Named Entity Recognition in the Molecular Biology Domain

Tuangthong Wattarujeekrit and Nigel Collier

National Institute of Informatics, 2-1-2 Hitotsubashi,
Chiyoda-ku, Tokyo, Japan 101-8430
tuangthong@grad.nii.ac.jp, collier@nii.ac.jp

Abstract. In this paper, the semantic relationships between a predicate and its arguments in terms of semantic roles are employed to improve lexical-based named entity recognition (NER) in the molecular biology domain. The semantic roles were realized in various sets of syntactic features used by a machine learning model to explore what should be the efficient way in allowing this knowledge to provide the highest positive effect on the NER. The empirical results show that the best feature set consists of *predicate's surface form*, *predicate's lemma*, *voice*, and the united feature of *subject-object head's lemma* and *transitive-intransitive sense*. The performance improvement from using these features indicates the advantage of the predicate-argument semantic knowledge on NER. There are still rooms to enhance NER by using this semantic knowledge (e.g. to employ other semantic roles besides *agent* and *theme* and to extend the rules for efficient identification of an argument's boundary).

1 Introduction

Named entity recognition (NER) is the task aiming to identify and categorize entities appearing in text. According to the Message Understanding Conferences (MUCs) [1], it is the lowest level in the task hierarchy of Information Extraction (IE) system. The entities to be recognized in the newswire domain include persons, organizations, locations, email addresses, and so on, whereas in the molecular biology domain, molecular entities such as genes, proteins, small molecules, chemical molecules, tissues, etc. need to be recognized. Not only is NER an important component of molecular biology IE to reach the goal of discovering biological pathways, but it is also beneficial to other applications of biological text mining. For instance, document retrieval where a relevant subset of documents are obtained [2] and document clustering where similar documents are grouped together [3]. For example, after NER has been used to process the sentence "*Cytokines bind to hematopoietin receptors and activate JAK kinases*", the fact that *Cytokines*, *hematopoietin receptors* and *JAK kinases* are referred to three different types of protein would be extracted. The different focus among researches gives variety to the granularity of concept classes to be distinguished. For example, to work with the GENIA ontology, 36 biologically nominal categories needed to be grouped [4].

Although, NER in the molecular biology domain has received wide scale attention by many researchers for nearly a decade, the overall performance is still far from

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 267 – 280, 2005.

human's capability [5-12]. As can be seen from the most recently shared-task of NER in the molecular biology domain (JNLPBA-2004), the best performance is only 72.6 for F-measure [9]. Contrastingly, the accuracy in general news-based NER is about 96% in MUC-6 [1] which is at near human levels of performance. This lag should mainly be due to the lack of naming convention[1] which leads to several sources of difficulties for NER. This work aims to handle two main difficulties as follows. First, the difficulty results from terminological variations i.e. molecular names may be formed by using a standard English word (e.g. "*light*", "*map*", "*complement*") or using an amino acid sequence (e.g. "*amino acids [aa] 1 to* 25") or using alpha numeric (e.g. "*9-cis retinoic acid*"). Second, the difficulty is from polysemy which is the ambiguity of a name that can refer to two or more different entities. Polysemy is classified into two cases: homonymy and systematic polysemy. Homonymy relates to the ambiguity of a name referring to unrelated meanings or objects (e.g. the term "*cat*" can refer to "*choline acetyltransferase protein*" and "*catalase gene*"). Systematic polysemy relates to the ambiguity of a name referring to the objects which systematically relate to each other (e.g. the term "*BCL-6*" can refer to "*B-cell CLL/lymphoma 6 gene*" and its protein product). These difficulties are expected to increase when we scale-up NER from an abstract to full text. Thus, most molecular NER systems now take place on MEDLINE abstracts.

In this paper, we argue that to overcome the limits in what can be achieved by existing NER systems traditionally based on lexical features and context features derived from neighboring words [7, 10-12], deeper knowledge such a predicate-argument relationship should be taken into account. This hypothesis is motivated by the basic observation that events are realized as predicates[2] and their participating named entities (NEs) as the predicates' arguments. The semantic role each argument plays in the event should impose type restrictions on the entity within the argument. The investigation of how to efficiently transform the knowledge of predicate-argument relations into features of training data for our NER system using a machine learning approach is the main focus in this work.

The paper is organized as follows. Section 2 discusses how predicate-argument relation is useful to NER and how other researchers have taken efforts to apply this knowledge. Section 3 outlines the transformation of predicate-argument relations into our machine learning features. Section 4 shows experimental results and the analysis on the results. Section 5 discusses concerning impediments to high performance improvement. Finally, Section 6 summarizes the conclusion.

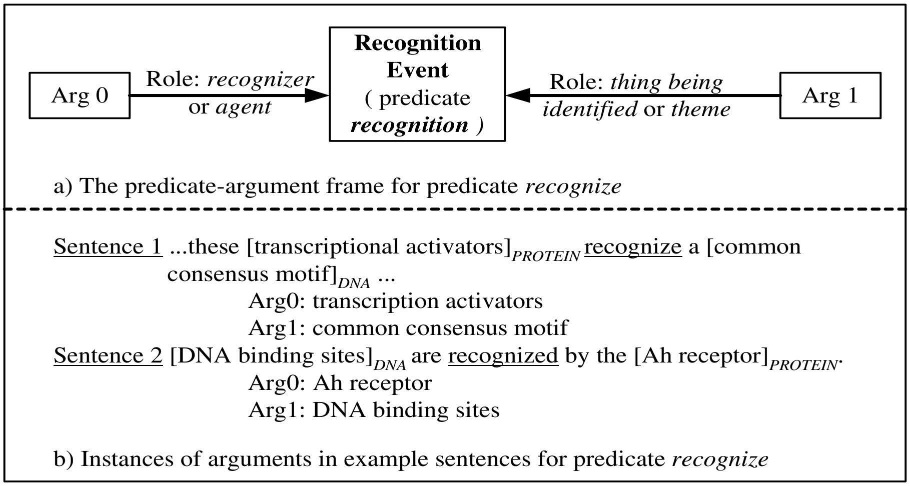
2 Predicate-Argument Relation and Biological NER

A frame of predicate-argument structure (PAS) represents a set of semantic relationships in terms of the specified role each argument plays in the event indicated by a

[1] Some efforts have been shown to standardize in naming biological entity (e.g. Guidelines of Human Gene Nomenclature, Drosophila Gene Nomenclature, etc., however many biologists often do not follow the recommended nomenclature.

[2] Hence, a predicate refers to a verb which can exist in a sentence in its verbal form (e.g. infinitive – *to activate*, present simple – *activate* or *activates*, past simple – *activated*, present or past participial – *activating* or *activated*), or its nominal form (e.g. *activation*).

predicate. For example, the predicate-argument frame of the predicate *recognize* which is used to express the recognition event in the molecular biology domain would be as Fig. 1(a). Thus, deeper knowledge than surface syntax of sentence 1 and 2 can be obtained as shown in Fig. 1(b). That is the occurrence of a recognition event would be participated by two participants (i.e. Arg0 and Arg1). The first argument (Arg0) has a relationship to the predicate *recognize* as a *recognizer* or *agent* of the event and the second argument (Arg1) plays role as *thing being identified* or *theme* in the event. Sentence 1 shows the usage of predicate *recognize* in active voice. The sentence's surface subject which is "*transcriptional activators*" plays role as *agent* and its surface object "*common consensus motif*" plays role as *theme*. On the contrary, a surface subject of sentence 2 which is "*DNA binding sites*" plays role as *theme* and a surface object "*Ah receptor*" plays role as *agent* as the predicate *recognize* is used in *passive voice*.


Arg 0
Role: recognizer or agent
Recognition Event (predicate recognition)
Role: thing being identified or theme
Arg 1

a) The predicate-argument frame for predicate *recognize*

Sentence 1 ...these [transcriptional activators]$_{PROTEIN}$ recognize a [common consensus motif]$_{DNA}$ ...
Arg0: transcription activators
Arg1: common consensus motif

Sentence 2 [DNA binding sites]$_{DNA}$ are recognized by the [Ah receptor]$_{PROTEIN}$.
Arg0: Ah receptor
Arg1: DNA binding sites

b) Instances of arguments in example sentences for predicate *recognize*

Fig. 1. The semantic relationships between predicate *recognize* and its argument

As can be noticed from Fig. 1, the argument playing role as *agent* belongs to class *PROTEIN* in both sentences. Similarly, the argument with semantic roles of *theme* belongs to class *DNA*. This restriction of NE-types corresponding to arguments' semantic roles is a key concept to employ semantic relations in PAS for enhancing molecular NER system.[3] As the NER system used in this work is based on Support Vector Machines (SVMs) [13], this predicate-argument relationship knowledge is required in the form of machine learning features.

Recently, due to the ability of PAS to straightforwardly represent the biological event, this knowledge has been used mostly as a reference frame to extract instances of biological events from text, e.g. the protein-protein interaction event [14-17]. To our knowledge, two previous works have shown the efforts to employ this knowledge

[3] The empirical evidence observed on GENIA V3.02 corpus (http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/) shows that the frequency of occurrence for *PROTEIN* to be *agent* in an recognition event is about 53% and for *DNA* to be *theme* is about 26%.

for NER in the molecular biology domain [6, 8]. In the first approach [6], the verb complementation patterns between each verb and the arguments which their concept classes are known have been automatically learnt by using an iterative reasoning process based on a partial order relation induced by the domain-specific ontology. Then, an unknown class term will be classified to the potential class based on the similarity measure between this new term's verb complementation patterns and the pre-analyzed known class term. This method still gets low performance to classify terms related to the small set of verbs that were studied (i.e. F-measure = 40.68%, 26.28%, 21.85%, and 19.69% for *bind*, *inhibit*, *interact*, and *mediate* respectively). In the second approach [8], a set of verbs, such as *inhibit*, *express*, *bind*, and *activate* has been set as binary features in HMM-based model. Unexpectedly, the overall F-measure has decreased by 1.8. One possible explanation for this result is that it could be due to the impractical way to represent predicate-argument relations in the model. The verb features represented only the knowledge that the verb exists in the context of the term or not.

In this paper, we explore an efficient way to exploit the semantic relations between predicate and its argument for improving SVM-based NER system.

3 Our Method

Our SVM-based NER system develops from the learning model of Takeuchi and Collier [7] in which the Tiny SVM[4] with the multi-class strategy of one-against-one was used. The context window was set to ± 1 providing features for the previous word, current word, and next word. Also, the two previous class assignments were taken into the model. The training data used in our system is in a form of a column formatted table of features with the NE classes provided in IOB2 format[5]. We form 6 sets of features (i.e. the Model 1 – Model 6) to be trained by SVMs. Model 1 contains only lexical-based features proposed in earlier studies to reduce known problems of ambiguity for term recognition. This model is used as a base model to be compared with the Model 2-6 in which predicate-argument related features are included in addition to lexical-based features. Thus, the significance of the semantic relationships represented in PAS to NER system can be evaluated. In order to evaluate the efficiency of different ways to convert this semantic knowledge into features of input data, Model 3, 4, 5, and 6 will be compared to the Model 2. How each feature set is derived and what thought is underlying the forming of it will be explained in section 3.3.

3.1 Data Set

The GENIA corpus V3.02, the largest annotated corpus in the molecular biology domain available to public, is used as our data set of the NE tagged text. As the predicate-argument relationship is a specific characteristic for each individual predicate, we decide to explore the influences of features derived from the knowledge of predicate-argument relation separately for each predicate. In this paper, we mainly focus to

[4] The Tiny SVM package is available from http://chasen.org/~taku/software/TinySVM/.

[5] IOB2 format is a standard format for word-chunk. The tag "O" is given to words outside a chunk, "B-*k*" to the first word in a chunk of type *k*, and "I-*k*" to the remaining words.

a predicate in verbal form, thus a collection for each predicate will be retrieved from GENIA by using the criteria that the relevant sentences must contain a focus predicate in verbal form at least once. With regard to the classes of NE used in evaluation, we follow the JNLPBA-2004 shared task [9] to use the conflated set of classes consisting of 5 classes: protein, DNA, RNA, cell line, and cell type.

3.2 Selection of Predicates to Be Explored

We started selecting predicates by gathering predicates used in earlier works to capture biological events [14-16] and predicates used in our previous work to construct the PASBio[6] resource [18]. Most predicates from the 44 predicates which have been gathered are found rarely in the GENIA corpus. In order to avoid having too small set of training data, we filtered out predicates containing less than 100 examples[7]. This filtering process results in a set of 19 predicates in which *bind* has a biggest and *alter* has a smallest volumes of training data, i.e. 825 and 102 examples respectively.

Table 1. The proportion of *agent* and *theme* arguments to 5 classes of NEs

Predicate		Group 1		Group 2		Group 3	
		Agent and theme with High NE%		Agent and Theme with Low NE%		Only Agent or Theme with High NE%	
		encode	recognize	block	lead	regulate	associate
Agent Argument	Total Agent	228	113	209	241	381	39
	Protein%	03.51	53.10	28.71	06.64	54.33	41.03
	DNA%	47.81	00.00	02.39	00.41	08.92	00.00
	RNA%	04.82	00.00	00.00	00.00	00.00	00.00
	Cell line%	00.44	07.08	00.48	01.24	00.00	05.13
	Cell type%	00.44	14.16	00.48	00.83	01.05	05.13
	Total NE%	57.02	74.34	32.06	09.13	64.30	51.29
	Non-NE%	42.98	25.66	67.94	90.87	35.70	48.71
Theme Argument	Total Theme	234	94	234	296	482	614
	Protein%	66.67	25.53	11.54	02.36	10.17	16.61
	DNA%	00.85	25.53	01.71	00.68	10.17	05.05
	RNA%	00.85	00.00	00.85	00.00	00.21	00.00
	Cell line%	00.00	00.00	00.00	00.00	00.41	00.49
	Cell type%	00.00	00.00	01.28	00.34	00.41	01.95
	Total NE%	68.37	51.06	15.38	03.38	21.37	24.10
	Non-NE%	31.63	48.94	84.62	96.62	78.63	75.90

Due to our intuition that the proportion of belonging to a NE class of an *agent* argument and a *theme* argument[8] should be a key impact to the performance of NER system when predicate-argument related features are applied, we selected 6 predicates

[6] PASBio resource contains frames of predicate-argument structure analyzed from the literatures in MB domain. Available online at http://research.nii.ac.jp/~collier/projects/PASBio/.

[7] The number of examples is a number of clauses containing a particular predicate. In a sentence, it is possible to have more than one clause related to the predicate in focus.

[8] The *agent* argument refers to the argument which has syntactic role as *subject* in the case of active voice and refers to the argument having syntactic role as *object* introduced by the preposition "by" in the case of passive voice. The *theme* argument refers to the argument which has syntactic role as *object* in the case of active voice and refers to the argument having syntactic role as *subject* in the case of passive voice.

from the total 19 predicates to be the representative predicates of the 3 groups as follows. First, the predicates *encode* and *recognize* were selected to be the representatives for a group of predicates having arguments both *agent* and *theme* with higher possibility to belong to a NE class than non-NE class. Second, the predicates *block* and *lead* were selected for a group of predicates having arguments both *agent* and *theme* with lower possibility to belong to a NE class than non-NE class. Third, the predicates *regulate* and *associate* were selected for a group of predicate having arguments either *agent* or *theme* with higher possibility to belong to a NE class than non-NE. Table 1 shows the proportion of the arguments of these representative predicates to 5 classes of NEs.

Moreover, if the number of examples for predicates from each group is not in balance, it could be difficult to compare their results. The intention to balance the number of examples of predicates to be investigated had been applied for selecting these representative predicates as well. More precisely, these 6 predicates were selected because they also conform to the condition that they have the numbers of examples nearly the average value for the total 19 predicates.

3.3 Derivation of Feature Sets

The Conexor FDG parser [19] which is widely used and is considered to be a state-of-the-art commercial parser is used to parse our NE tagged text. In addition to each word's morphological information (i.e. surface form and lemma form) and lexical category (i.e. part-of-speech), this parser also provides functional dependency relations between words which is one of a key syntactic information for acquiring semantic relationships between a predicate and its arguments. These parsing results are used to derive a set of features used in the Model 1-6 as follows.

Model 1. This model composes of 6 features widely recognized as important for NER task. These features include *surface word*, *lemma form*, *head word of noun phrase*, *part-of-speech*, *orthographic feature*, and *phrase-chunk*. This model is named lexical-based model as it is based mainly on lexical information. As stated before, this model is used as a base model for evaluating the importance of the semantic knowledge represented in PAS to NER system.

Model 2. This model contains all lexical-based features used in the Model 1, with additional set of features constituted from syntactic information to represent arguments' semantic roles). These supplementary features consist of *predicate surface form*, *predicate lemma*, *voice* and *surface syntactic role*. The *voice* feature is used to distinguish between *active* and *passive* voice of the predicate. The tag "*ACT*" represents *active* voice and "*PAS*" represents *passive* voice. The *surface syntactic role* feature describes syntactic functions (i.e. *surface subject* or *surface object*) of the head word of a noun-phrase which is bound as the predicate's argument. Tags used are "*SSUBJ*" for *surface subject* and "*SOBJ*" for *surface object* which is found as direct object. Moreover, the tag "*PCOMP*" used for *surface object* which is found as a prepositional complement. For instance, from sentences "A binds B." and "A binds to B.", "A" will be tagged with "*SSUBJ*" in both sentences but "B" will be tagged with "*SOBJ*" for the former sentence and "*PCOMP*" for the latter. The procedures used to identify the argument's boundary are illustrated in section 3.4. The semantic roles of arguments can be determined partially from a combination of the 4 additional features used in this model. Only if both *surface subject* and *surface object* co-occur with a target verb, the argument with syntactic function as *subject* and the argument with syntactic function as *object* will be confidently concluded that they semantically plays role as *agent* and *theme* respectively in case of *active* voice and vice versa in case of *passive* voice. The correct determination of semantic role would lead to the correct NE classification; underlying our hypothesis that semantic relationships in PAS (arguments' semantic roles) for each predicate confine classes of NEs participating the event indicated by the predicate. However, as the arguments with the same semantic role possibly belong to different NE classes, the lexical-based features and semantic relationships are required altogether to solve this ambiguity. This model is a PAS-based model which will be extended to the Model 3-6 by adding features of several kinds of syntactic information in order to decrease the ambiguity in determining semantic roles.

Model 3. Path feature representing the syntactic path from the subject argument to the related predicate and from the related predicate to the object argument is added to all features used in Model 2. The path is derived from the flat structure of dependency tree resulting from the parser. For example, the path between the subject constituent and the predicate is "NP_VP_ADVP_VP" and the path between the object constituent and the predicate is "VP_PP_NP" for the sentence "[Increased cytokine secretion]$_{\text{NP}}$ [was]$_{\text{VP}}$ [specifically]$_{\text{ADVP}}$ [inhibited]$_{\text{VP}}$ [by]$_{\text{PP}}$ [G1]$_{\text{NP}}$".

Model 4. A feature representing a pair of subject and object's heads is added to the Model 2 instead of path feature. This feature is designed following the intuition that a NE class of an agent should restrict a possible type of a NE playing role as theme and vice versa. The using of a subject-object head pair in lemma form would help to reduce data sparseness problem compared to the using in surface form. For the sentence in Fig. 2, the subject-object head feature will be "compound_complex".

Model 5. This model augments the Model 2 with a feature representing if a predicate is used in transitive or intransitive sense. For each surface subject's constituent, a tag "fobj" is set if the surface object is found in the current clause. A tag "O" is set if the surface object is not found. However, this feature helps just in part to correctly determine transitive or intransitive sense implicit in the usage of a predicate. It is due to the object argument can be omit in a clause although a predicate is used in transitive sense. For instance, the predicate "eat" is used in transitive sense without mentioning an object in the sentence "Yesterday, John ate at ABC restaurant".

Model 6. This model is considered as a joining of the Model 4 and the Model 5. A pair of subject and object's heads is used to be assigned to a column of transitive-intransitive feature instead of "fobj" when the object is found in the clause.

The lexical-based features used in Model 1 will be given to every word or token in a sentence. Contrastingly, the PAS-related features proposed in Model 2-6 will be assigned to only the constituents bound as the arguments having syntactic function as *surface subject* and *surface object* of the focus predicate. How to identify the boundary of these constituents is as follows.

3.4 Sub-Structure Recognition

The sub-structure recognition is the process to identify the tokens that constitute arguments of predicates. In our study, we have focused mainly on a predicate in verbal form but not nominal form. Therefore, for a predicate such as *activate*, the surface forms of this predicate to be analyzed include *activate*, *activates*, *activated*, and *activating*¸ but not *activation*. Furthermore, only an argument corresponding to the syntactic relation of either subject or object is bound in this study. At present, there is a lack of practical semantic role labeling systems to identify arguments of a predicate, especially for the molecular biology domain. Thus, this study which is to investigate the constitution of semantic relationship between predicate and its arguments simplifies its scope to arguments as grammatical subject or object.

C. 1	C. 2	C. 3	C. 4	C. 5	C. 6	C. 7	
Word No.	**Surface Form**	**Lemma Form**	**Syntactic Relation**	**Functional Tag**	**Surface Syntac-tic**	**Part-of-Speech**	
1	Both	Both	det:>2	@DN>	%>N	DET -	
2	compounds	compound	subj:>3	@SUBJ	%NH	N NOM_PL	**Subject**
3	altered	Altered	main:>0	@+FMAINV	%VA	V PAST	**Verb**
4	the	The	det:>7	@DN>	%>N	DET -	
5	NFAT-1	NFAT-1	attr:>6	@A>	%>N	N NOM_SG	**Object**
6	transcrip-tional	transcrip-tional	attr:>7	@A>	%>N	A ABS	
7	complex	Complex	obj:>3	@OBJ	%NH	A ABS	
8	,	,					
9	causing	Causing	ha:>3	@-FMAINV	%VA	V ING	
10	its	Its	attr:>11	@A>	%>N	PRON GEN_SG 3	
11	retarded	retarded	attr:>12	@A>	%>N	A ABS	
12	mobility	Mobility	obj:>9	@OBJ	%NH	N NOM_SG	
13	on	On	loc:>9	@ADVL	%EH	PREP -	
14	gels	Gel	pcomp:>13	@<P	%NH	N NOM_PL	
15	.	.					

Fig. 2. Boundaries of surface subject and object of the verb *alter* recognized by the system (*thick squares*) using the FDG parsing result of a sentence "Both compounds altered the NFAT-1 transcriptional complex, causing its retarded mobility on gels"

The algorithm used to find a subject constituent and an object constituent of each predicate is based mainly on the functional dependency relations between words obtained from the parser as shown in Fig. 2. It comprises of several steps as follows. First, find a position of target predicate which must be in a verbal form. Second, interpret the verb's voice by checking at the column *Surface Syntactic* (Fig. 2, C. 6) of the verb token (Word No. 3). If it is *%VA*, the verb is an active verb. On the other hand, if it is *%VP*, the verb is a passive verb. Third, find a token functioning as a subject or object of the target verb by traversing through syntactic relations given by the parser (Fig. 2, C. 4). Basically, the system will traverse up until *subj:>#* is found in case of subject and traverse down until *obj:>#* is found in case of object.[9] From Fig. 2, the token *compounds* is found to have subject relation to the verb *alter* and the token *complex* is found to be an object. Subsequent to founding the head of subject or object, the full boundary of a subject or an object is identified by propagating to the premodifiers of a noun which is a subject head or an object head. These premodifiers will have @*A*> at the column *Functional Tag* in parsing data (Fig. 2, C. 5). All modifiers except determiners are included in surface subject or surface object boundary as shown in Fig. 2 that *NFAT-1* and *transcriptional* are included but *the* is not included in the boundary of surface object containing *complex* as the object head. A determiner is not included into both boundary of object and subject because any determiners never ever are parts of the biological terms. This rule not to include a determiner is also used by Rindflesch and colleagues to extract binding relationships [17].

To look for *subj:>#* or *obj:>#*, at the column *Syntactic Relation* (Fig. 2, C. 4), to get a subject head or an object head is practical for a simple clause. In some cases, a token holding *subj:>#* or *obj:>#* is not found as a subject head or an object head has a direct dependency relation to another token but not to a target verb. The more complex criterion needs to be processed to recover the relations between a subject and an object to the target verb. These cases are as follows: 1) an auxiliary verb (e.g. be, do, have, etc.) or a verb phrase functioning similar to auxiliary verb (e.g. play a role in, is required to, have been shown to, etc.) precedes a target verb, 2) a target verb shares its subject or object with other verbs, 3) a target verb is a main verb in a subordinate clause of which the relative pronoun presents as the subject, and 4) an object of a target verb is introduced by a preposition following a target verb[10].

4 Experimental Results and Analysis

All results reported here are given as F1-scores calculated using 10-fold cross validation. F1-score is defined as *F1= (2PR)/(P+R)* where *P*, called as *Precision*, is the ratio of the number of correctly found NE chunks to the number of found NE chunks and *R*, called as *Recall*, is the ratio of the number of correctly found NE chunks to the number of true NE chunks.

The results of 6 predicates using the feature sets from the Models 1-6 are shown in Table 2. In each column, the F1-score of a corresponding predicate is given for Model 1 (Lexical-based model), Model 2 (PAS-related model), Model 3 (the Model 2 added with Path feature), Model 4 (the Model 2 added with Pair of subject and object's heads feature), Model 5 (the Model 2 added with Transitive/Intransitive feature) and the Model 6 (the Model 4 is embodied into the Model 5). For each predicate, the higher F1-scores from the models which outperform the Model 1 are shown in *bold* number. The models with *bold* number indicate the positive effect of PAS-related features to NER. Moreover, if the F1-scores in any models among Models 3-6 are

[9] Hence, the symbol # refers to the word number of the target verb.

[10] Due to the space limitation, the details of the extended criterion for these complicated cases to identify the boundaries of subject and object arguments cannot be explained here.

higher than in Model 2, the scores will be highlighted with *gray* background. This helps to notice which PAS-related feature in addition to features used in PAS-based model (Model 2) has capability to increase positive effect of semantic relations between predicate and its arguments.

Table 2. F1-scores of the 6 representative predicates trained with features in Models 1-6

Predicates / Model	Group 1		Group 2		Group 3	
	Agent and Theme with High NE%		Agent and Theme with Low NE%		Only Agent or Theme with High NE%	
	encode (265)	recognize (121)	block (270)	lead (288)	regulate (525)	associate (377)
Model 1 (*Lexical-based*)	56.60	47.24	51.19	57.01	61.87	52.09
Model 2 (*PAS-based*)	**57.56**	**49.39**	**51.47**	**57.40**	60.48	51.48
Model 3 (*Path*)	**58.38**	**48.47**	**52.23**	56.70	60.13	51.29
Model 4 (*Head Pair*)	**57.16**	**49.54**	**51.85**	**57.12**	60.72	50.43
Model 5 (*Trans/Intrans*)	**57.69**	**49.16**	**52.02**	**57.53**	60.01	51.40
Model 6 (*M4+M5*)	**57.64**	**49.39**	**51.95**	**57.49**	60.37	50.97

As can be observed from Table 2, the simple representation of PAS-related knowledge such in Model 2 improve the performance for all predicates except the predicates *regulate* and *associate* which have only *agent* or *theme* argument with higher possibility to belong to a NE class than non-NE. Moreover, these Group 3's predicates do not show any improvement in other models using PAS-related features (Model 3-6) compared to the lexical-based model (Model 1). Therefore, they will not be covered in the following discussion of how the extra PAS-related features used in Models 3-6 help to improve the performance of PAS-based features used in Model 2.

With regard to Path feature (Model 3), the performance is improved for only the model training on data set of predicate *encode* and *block*. Empirically, one reason we found for this is the surface subject and surface object of these two predicates are located close to the predicate in most of the cases. For example, the path patterns between arguments and the predicate *encode* of "…[proteins]$_{\text{NP}}$ [encoded]$_{\text{VP}}$ [by]$_{\text{ADVP}}$ [these two latter genes]$_{\text{NP}}$…" are "NP_VP" for the subject argument and "VP_ADVP_NP" for the object argument. Due to short path patterns, so the path patterns can be generalized throughout the data sets. On the contrary, long path patterns are mostly found in the samples of other predicates (i.e. *recognize* and *lead*). For example, from the sentence "[Control peptides]$_{\text{NP}}$ [corresponding]$_{\text{VP}}$ [to]$_{\text{ADVP}}$ [the normal pml]$_{\text{NP}}$ [and]$_{\text{O}}$ [RAR alpha proteins]$_{\text{NP}}$ [were]$_{\text{VP}}$ [not]$_{\text{ADVP}}$ [recognized]$_{\text{VP}}$.", the path from the subject argument "Control peptides" to the predicate *recognize* is "NP_VP_ADVP_NP_O_NP_VP_ADVP_VP". This long path pattern would causes data sparseness problems for the path feature.

The next feature, the Head Pair feature, does not show its usefulness for predicates *encode* and *lead*. The reason for the predicate *lead* is that its arguments both as *agent* and *theme* are prone to be non-NE rather than to belong to NE class, thus the pair of its arguments' head words can have many variants. It causes this feature ineffective to constrain NE functioning as subject with NE functioning as object and vice versa. In case of predicate *encode*, although both arguments of it are prone to belong to NE classes rather than to be non-NE, the Head Pair feature does not show its positive effect. As the predicate *encode* used in the molecular biology domain has its specific meaning to describe relationships between genes and gene products, the head pair of arguments for this predicate is mostly found as *gene_protein*. Therefore, this feature contains too general information to be helpful for *encode*.

In case of Transitive/Intransitive feature, we believe that this feature should be useful to improve performances of all predicates. This feature is important to correctly interpret semantic role of an argument. For instance, the subject in the sentence "John broke the window" has the semantic role as *agent* but the subject in the sentence "The window broke" has semantic role as *theme*. These two sentences illustrate that to know only syntactic function as subject or object cannot have a correct determination on semantic role. The difference between these two sentences is that the predicate *break* is used in transitive sense in the former sentence and intransitive sense in the latter. Therefore, to give information stating if the object is found in a sentence or not would help to some extent to imply sense in which the predicate is used. The performance of the model having this feature (Model 5) should outperform the PAS-based model (Model 2). However, the performance for *recognize* has decreased when this feature is applied. From our analysis, the problem originates from parsing error of failing to provide syntactic relations between words. For instance, the FDG parser fails to give the constituent "DNA binding sites" syntactic relation as the object of "recognizes" in the sentence "The Ah receptor recognizes DNA binding sites for the B cell transcription factor" This causes subsequent problem to the Transitive/Intransitive feature, i.e. this feature is set to "O" to represent that the predicate *recognize* is used in intransitive sense, whereas it does not. This incomplete parsing result accounts for decreasing F1-score of *recognize* when using the Transitive/Intransitive feature (Model 5) compared to when not using it (Model 2).

In order to evaluate the contribution of PAS-related features from different models, the average F1-score from each PAS-related model (Model 2-6) is compared to the average F1-score of the lexical-based model (Model 1). Without considering the mix model (Model 6), the results show that the Transitive/Intransitive feature (Model 5) gives the highest contribution as expected. Some more improvement can be obtained in Model 6 when the Head Pair feature (Model 4) is embedded in the Transitive/Intransitive feature (Model 5). Thus far, the Model 6 is considered to be the best model in this work with the improvement, on average, in F1-score of 1.11 as shown in Table 3. Furthermore, each predicate reflects the benefit from using PAS-related features in different levels of improvement, listed in descending order as *recognize*, *encode*, *block*, and *lead*.

Table 3. The improvement in F1-scores of Model 6 (the best of PAS-related model) compared to Model 1 (the lexical-based model)

Predicates	Number of Examples	Model 1 (*Lexical-based*)	Model 6 (*M4+ M5*)	Improvement
Encode	265	56.60	57.64	1.04
Recognize	121	47.24	49.39	2.15
block	270	51.19	51.95	0.76
lead	288	57.01	57.49	0.48
Average of Improvement				1.11

5 Discussions

In Table 3, the experimental results have shown that the PAS-related features make only small progress in NER. However, it is not because semantic relationship between predicate and its argument is not an important knowledge to improve lexical-based NER. The incorrect identification of an argument boundary is an impediment for the system to acquire the actual performance improvement. This impediment is mainly caused by a failing of parser to provide syntactic relation information between tokens. One of its examples has already been shown in the previous section to explain why the Transitive/Intransitive feature degrades the performance of *recognize*. To investigate the contribution of PAS-related features without the impact from parsing error, the arguments *agent* and *theme* are identified manually on training examples of predicates *recognize* and *encode* (100 examples for each predicate). These two predicates are selected for this experiment because they obtain higher performance improvement than other predicates. The 2 sets of training data are trained by using features in Model 1 and Model 6 to calculate the performance improvement. The predicate *encode* obtains performance improvement of 2.40 from training on only 100 manual-examples (about 38% of parsing-examples) [11]. This performance improvement is about 2 times of what obtained from 265 parsing-examples (Table 3). In case of predicate *recognize*, from training on 100 manual-examples, the performance improvement increase to 6.12 which is about 3 times of what is obtained from 121 parsing-examples. The size of manual-examples of *recognize* is nearly equal to the parsing-examples' size, thus it can be implied that the parsing error can decrease the performance improvement at least 3 times.

In addition to the parsing error, the more complex rule to identify an argument boundary is required for some specific cases. For example, the constituent "multiple isotypes" in the sentence "T cells express multiple isotypes of protein kinase C" will be bounded to be *theme* argument of predicate *express* after the general algorithm for sub-structure recognition is applied. However, the real argument playing semantic role as *theme* which is related to NE-type protein is the constituent "protein kinase C". Therefore, a set of rules to distinguish between a quantifier (e.g. "multiple isotypes") and a real argument (e.g. "protein kinase C") is required. Moreover, a rule set to include or not to include an entity's abbreviation name (always mentioned in a bracket) in an argument boundary is required as well. For instance, in GENIA corpus V3.02 the constituent "cytokine receptor gamma chain (gamma c) gene" of a sentence "…cytokine receptor gamma (gamma c) gene encodes a component of …" is hand-annotated as one named entity, but the constituent "Sterol regulatory element (SRE)" of a sentence "….Sterol regulatory element (SRE) has been recognized …" is separated into two named entities (i.e. "Sterol regulatory element" and "SRE").

In order to allow semantic knowledge of predicate-argument relationship covering semantic roles of *agent* and *theme* to express its actual contribution, the sources of errors in identifying an argument boundary as mentioned above must be handled.

[11] Hence, the training examples are called manual-examples when argument boundaries are identified manually and are called parsing-examples when argument boundaries are identified automatically based on syntactic relation information given by the parser.

6 Conclusions

In this work, we have shown that the deeper knowledge of semantic relationship between a predicate and its argument is benefit for NER. The choice of syntactic features to represent the PAS semantic knowledge is the key issue underlying the efficient employment of this knowledge. So far, the best set of syntactic features consists of features *predicate's surface form*, *predicate's lemma*, *voice*, and the united feature of *subject-object head's lemma* and *transitive-intransitive sense*. The highest improvement is found from applying these features to the training examples of predicate *recognize*. Without parsing error which is one of the problems that can impede the contribution of the predicate-argument semantic knowledge to NER system, the highest improvement for *recognize* can reach to 6.12 F1-score.

Besides dealing with an argument's boundary identification discussed in this work, there are still rooms to enhance NER by using this PAS knowledge such as employing syntactic features to represent other semantic roles in addition to *agent* and *theme*.

References

1. DARPA. The 6th Message Understanding Conference. Columbia, Maryland (1995)
2. Stapley, B. J., Benoit, G.: Biobibliometrics: Information retrieval and visualization from co-occurrences of gene names in Medline abstracts. Pac. Symp. Biocomp.(2000) 529-540
3. Willett, R.: Recent trends in hierarchic document clustering: a critical review. Information Processing & Management. (1998) 25: 577
4. Ohta, T., Tateishi, Y., Kim, J. D.: The GENIA corpus: An annotated research abstract corpus in the molecular biology domain. HLT. (2002)
5. Fukuda, K., Tamura, A., Tsunoda, T., Takagi, T.: Toward information extraction: identifying protein names from biological papers. Pac. Symp. Biocomp. (1998) 707-718
6. Spasic, I., Nenadic, G., Ananiadou, S.: Using domain-Specific Verbs for Term Classification. The ACL Workshop on NLP in Biomed. (2003) 17-24
7. Takeuchi, K., Collier, N.: Use of Support Vector Machines in Extended Named Entity Recognition. CONLL. (2002) 119-125
8. Zhou, G., Su, J.: Exploring Deep Knowledge Resources in Biomedical Name Recognition. The Joint Workshop on NLP in Biomed. and its App. (JNLPBA). (2004) 84-87
9. Kim, J. D., Ohta, T., Tsuruoka, Y., Tateishi, Y., Collier, N.: Introduction to the Bio-Entity Task at JNLPBA. (2004) 70-75
10. Collier, N., Nobata, C., Tsujii, J.: Extracting the names of genes and gene products with a Hidden Markov Model. COLING. (2000) 201-207
11. Kazama, J., Makino, T., Ohta, Y., Tsujii, J.: Tuning Support Vector Machines for Biomedical Named Entity Recognition. The ACL Workshop on NLP in Biomed. (2002) 1-8
12. Lee, K. J., Hwang, Y. S., Rim, H. C.: Two-phase biomedical NE Recognition based on SVMs. The ACL Workshop on NLP in Biomed. (2003) 33-40
13. Vapnix, V. N.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1998)
14. Blaschke, C., Andrade, M. A., Ouzounis, C., Valencia, A.: Automatic extraction of biological information from scientific text: Protein-protein interactions. The Int. Conf. on Intelligent System Molecular Biology. (1999) 60-67

15. Ono, T., Hishigaki, H., Tanigami, A., Takagi, T. Automated extraction of information on protein-protein interactions from the biological literature. Bioinform. (2001) 17:155-161
16. Pustejovsky, J., Castano, J., Zhang, J.: Robust Relational parsing over Biomedical Literature: Extracting Inhibit Relations. Pac. Symp. Biocomput. (2002) 505-516
17. Rindflesch, T. C., Rajan, J. V., Hunter, L.: Extracting Molecular Binding Relationships from Biomedical Text. ANLP. (2000) 188-195
18. Wattarujeekrit, T., Shah, P., Collier, N.: PASBio: predicate-argument structures for event extraction in molecular biology. BMC Bioinformatics. (2004) 5: 155
19. Tapanainen, P., Jarvinen, T.: A non-projective dependency parser. ANLP. (1997) 64-71

Massive Biomedical Term Discovery

Joachim Wermter and Udo Hahn

Jena University,
Language and Information Engineering (Julie) Lab,
Fürstengraben 30
Jena, 07743, Germany
http://www.coling.uni-jena.de

Abstract. Most technical and scientific terms are comprised of complex, multi-word noun phrases but certainly not all noun phrases are technical or scientific terms. The distinction of specific terminology from common non-specific noun phrases can be based on the observation that terms reveal a much lesser degree of distributional variation than non-specific noun phrases. We formalize the limited paradigmatic modifiability of terms and, subsequently, test the corresponding algorithm on bigram, trigram and quadgram noun phrases extracted from a 104-million-word biomedical text corpus. Using an already existing and community-wide curated biomedical terminology as an evaluation gold standard, we show that our algorithm significantly outperforms standard term identification measures and, therefore, qualifies as a high-performant building block for any terminology identification system.

1 Introduction

With proliferating volumes of medical and biological text available, the need to extract and manage domain-specific terminologies has become increasingly relevant in the recent years. Most available terminological dictionaries, however, are still far from being complete, and what's worse, a constant stream of new terms enters via the ever-growing biomedical literature. Naturally, the costly and time-consuming nature of manually identifying new terminology from text calls for procedures which can automatically assist database curators in the task of assembling, updating and maintaining domain-specific controlled vocabularies. Whereas the recognition of single-word terms usually does not pose any particular challenges, the vast majority of biomedical or any other domain-specific terms typically consists of multi-word units,[1] which are, thus, much more difficult to recognize and extract. Moreover, although the need to assemble and extend technical and scientific terminologies is currently most pressing in the biomedical domain, virtually any (sub-)field of human research / expertise in which structured knowledge needs to be extracted from text collections calls for high-performance terminology identification methods.

[1] According to [1], more than 85% of domain-specific terms are multi-word units.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 281–293, 2005.

2 Related Work and Purpose

There have been many studies examining various methods to automatically extract scientific or technical terms from domain-specific corpora, such as from biomedical ones (see, e.g., [2], [3], [4], [5], [6] and [7]). Typically, approaches to multi-word term extraction collect term candidates from texts by making use of various degrees of linguistic filtering (e.g., part-of-speech tagging, phrase chunking etc.), through which candidates of various linguistic patterns are identified (e.g. *noun-noun*, *adjective-noun-noun* combinations etc.). These possible choices are then submitted to frequency- or statistical-based evidence measures (e.g., C-value [8]) which compute weights indicating to what degree a candidate qualifies as a terminological unit. While *term mining*, as a whole, is a complex process involving several other components (e.g., orthographic and morphological normalization, acronym detection, conflation of term variants, term context, term clustering, etc., see [6]), the measure which assigns a *termhood value* to a term candidate is an essential building block of any term identification system.

In multi-word automatic term recognition (ATR) the C-value approach [8, 9], which aims at improving the extraction of nested terms, has been one of the most widely used techniques in recent years. Other potential association measures are mutual information [10], and the battery of statistical and information-theoretic measures (t-test, log-likelihood, entropy) which is typically employed for the extraction of general-language collocations (see [11, 12]). While these measures have their statistical merits in terminology identification, it is interesting to note that they make little use of linguistic properties associated with terminological units.[2] However, such properties have proven to be helpful in the identification of general-language collocations [13]. Therefore, one may wonder whether there are linguistic features which may also be beneficial to ATR. One such feature we have identified is the *limited paradigmatic modifiability* of terms, which will be described in detail in Subsection 3.3.

The purpose of our study is to present a novel term recognition measure which directly incorporates this linguistic criterion, and in evaluating it against some of the standard procedures, we show that it substantially outperforms them on the task of term extraction from the biomedical literature.

3 Methods and Experiments

3.1 Construction and Statistics of the Training Set

We collected a biomedical training corpus of approximately 513,000 MEDLINE abstracts using the following MESH-terms query: *transcription factors*, *blood cells* and *human*.[3] We then annotated this 104-million-word corpus with the GENIA part-of-

[2] One notable exception is the C-value method which incorporates a term's likelihood of being nested in other multi-word units.

[3] MEDLINE is a large biomedical bibliographic database (see `http://www.ncbi.nlm.nih.gov`). For information retrieval purposes, all its abstracts are indexed with a controlled indexing vocabulary, viz. MESH. Our query is aimed at the molecular biology domain, with the publication period from 1978 to 2004.

speech tagger[4] and identified noun phrases (NPs) with the YAMCHA-Chunker [14]. In this study, we restricted ourselves to NP recognition (i.e., determining the extension of a noun phrase but refraining from assigning any internal constituent structure to that phrase), because the vast majority of technical or scientific terminology (and terms in general) is contained within noun phrases [15]. We filtered out a number of stop words (i.e., determiners, pronouns, measure symbols etc.) and also ignored noun phrases with coordination markers (e.g., *and*, *or* etc.).[5]

Table 1. Frequency distribution for noun phrase term candidate tokens and types for our 104-million-word MEDLINE text corpus

n-gram length	cut-off	NP term candidates tokens	types
bigrams	no	5,920,018	1,055,820
	$c \geq 10$	4,185,427	67,308
trigrams	no	3,110,786	1,655,440
	$c \geq 8$	1,053,651	31,017
quadgrams	no	1,686,745	1,356,547
	$c \geq 6$	222,255	10,838

In order to obtain our term candidate sets (see Table 1), we counted the frequency of occurrence of noun phrases in our training corpus and categorized them according to their length. For this study, we restricted ourselves to noun phrases of length 2 (word bigrams), length 3 (word trigrams) and length 4 (word quadgrams). Morphological normalization of term candidates has shown to be beneficial for ATR [9]. We thus normalized the nominal head of each noun phrase (typically the rightmost noun in English) via the full-form UMLS SPECIALIST LEXICON [16], a large repository of both general-language and domain-specific (medical) vocabulary. To eliminate noisy low-frequency data, we set different frequency cut-off thresholds c for the bigram, trigram and quadgram candidate sets and only considered candidates above these thresholds (see Table 1).

3.2 Evaluating Terminology Extraction Algorithms

(Domain-specific) terms are usually referred to as the linguistic surface manifestation of (domain-specific) concepts. Typically, terminology extraction studies evaluate the goodness of their algorithms by having their ranked output examined by so-called *domain experts* who identify the true positives among the ranked candidates. There are several problems with such an approach. First, very often only one such expert is consulted and so inter-annotator agreement is not accounted for (e.g. in the studies of [8],

[4] `http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/postagger/`

[5] Of course, terms can also be contained within coordinative structures (e.g. *B and T cell*). However, analyzing their inherent ambiguity is a complex syntactic operation, with a comparatively marginal benefit for ATR [9].

[4]). Furthermore, what constitutes a relevant term for a particular domain may be rather difficult to decide – even for domain experts – if all they have in front of them is a list of candidates without any further context. Thus, rather than relying on direct human judgement in identifying true positives among a candidate set, a better solution may be to take already existing terminolgical resources, which have developed over years and have gone through various modifications and editions by expert committees. In this sense, the biomedical domain is an ideal test bed for evaluating the goodness of ATR algorithms because it hosts one of the most extensive and curated terminological resources, viz. the UMLS METATHESAURUS [17], and thus offers a well-established source of curated and agreed judgements about what constitutes a biomedical term.

Accordingly, for our purposes of evaluating the quality of different measures in recognizing multi-word terminology from the biomedical literature, we take every word bigram, trigram, and quadgram in our candidate sets to be a term (i.e., a true positive) if it was found in the 2004 UMLS METATHESAURUS.[6] For example, the word trigram "*long terminal repeat*" is listed as a term in one of the UMLS vocabularies, *viz.* MESH [18], whereas "*t cell response*" is not. Thus, among the 67,308 word bigram candidate types, 14,650 (21.8%) were identified as true terms; among the 31,017 word trigram candidate types, the number was 3,590 (11.6%), and for the 10,838 word quadgram types, 873 (8.1%) were identified as true terms.[7]

3.3 Paradigmatic Modifiability of Terms

For most standard association measures utilized for terminology extraction, frequency of occurrence of the term candidates either plays a major role (e.g., C-value) or at least has a significant impact concerning the degree of *termhood* assigned (e.g., t-test). However, frequency of occurrence in a training corpus may be misleading regarding the decision whether or not a multi-word expression is a term. For example, taking the two trigram multi-word expressions from the previous subsection, the non-term "*t cell response*" appears 2410 times in our 104-million-word MEDLINE corpus, whereas the term "*long terminal repeat*" (= long repeating sequences of DNA) only appears 434 times (see also Tables 2 and 3 below).

The linguistic property around which we built our measure of termhood is the *limited paradigmatic modifiability* of multi-word terminological units. For example, a trigram multi-word expression such as "*long terminal repeat*" contains three word/token slots in which slot 1 is filled by "*long*", slot 2 by "*terminal*" and slot 3 by "*repeat*". The *limited paradigmatic modifiability* of such a trigram is now defined by the probability with which one or more such slots *cannot* be filled by other tokens, i.e., the tendency not to let other words appear in particular slots. To arrive at the various combinatory possibilities that fill these slots, the standard combinatory formula without repetitions

[6] We excluded those UMLS source vocabularies that were definitely not deemed relevant for molecular biology, such as nursing and health care billing codes.

[7] As can be seen, not only does the number of candidate types drop with increasing n-gram length but also the proportion of true terms. In fact, their proportion drops more sharply than can actually be seen from the above data because the various cut-off thresholds have a leveling effect.

Table 2. $P\text{-}Mod$ and k-modifiabilities for $k = 1$ and $k = 2$ for the trigram term *long terminal repeat*

n-gram	freq	$P\text{-}Mod$ (k=1,2)	
long terminal repeat	434	0.03	
k slots	possible selections *sel*	freq	mod_{sel}
$k = 1$	k_1 terminal repeat	460	0.94
	long k_2 repeat	448	0.97
	long terminal k_3	436	0.995
		mod_1 =0.91	
$k = 2$	k_1 k_2 repeat	1831	0.23
	k_1 terminal k_3	1062	0.41
	long k_2 k_3	1371	0.32
		mod_2 =0.03	

Table 3. $P\text{-}Mod$ and k-modifiabilities for $k = 1$ and $k = 2$ for the trigram non-term *t cell response*

n-gram	freq	$P\text{-}Mod$ (k=1,2)	
t cell response	2410	0.00005	
k slots	possible selections *sel*	freq	mod_{sel}
$k = 1$	k_1 cell response	3248	0.74
	t k_2 response	2665	0.90
	t cell k_3	27424	0.09
		mod_1 =0.06	
$k = 2$	k_1 k_2 response	40143	0.06
	k_1 cell k_3	120056	0.02
	t k_2 k_3	34925	0.07
		mod_2 =0.00008	

can be used. For an n-gram (of size n) to select k slots (i.e., in an unordered selection) we define:

$$C(n, k) = \frac{n!}{k!(n-k)!} \tag{1}$$

For example, for $n = 3$ (a word trigram) and $k = 1$ and $k = 2$ slots, there are three possible selections for each k for "*long terminal repeat*" and for "*t cell response*" (see Tables 2 and 3). Here, k is actually a placeholder for any possible word/token (and its frequency) which fills this position in the training corpus.

Now, for a particular k ($1 \leq k \leq n$; n = length of n-gram), the frequency of each possible selection, sel, is determined. The paradigmatic modifiability for a particular selection sel is then defined by the n-gram's frequency scaled against the frequency of sel. As can be seen in Tables 2 and 3, a *lower* frequency induces a *more limited* paradigmatic modifiability for a particular sel (which is of course expressed as a higher probability value; see the column labeled mod_{sel} in the tables). Thus, with s being the number of distinct possible selections for a particular k, the *k-modifiability*, mod_k, of an n-gram can be derived as follows:

$$mod_k(n\text{-}gram) := \prod_{i=1}^{s} \frac{f(n\text{-}gram)}{f(sel_i, n\text{-}gram)} \tag{2}$$

Then, the *paradigmatic modifiability*, $P\text{-}Mod$, of an n-gram is the product of all its k-modifiabilities:[8]

$$P\text{-}Mod(n\text{-}gram) := \prod_{k=1}^{n} mod_k(n\text{-}gram) \tag{3}$$

Comparing the trigram $P\text{-}Mod$ values for $k = 1, 2$ in Tables 2 and 3, it can be seen that the term "*long terminal repeat*" gets a much higher weight than the non-term "*t cell response*", although their mere frequency values suggest the opposite. This is also reflected in the respective output list rank (see Subsection 4.1 for details) assigned to both trigrams by t-test and by our $P\text{-}Mod$ measure. While "*t cell response*" has rank 24 on the t-test output list (which has to be attributed to its high frequency), $P\text{-}Mod$ puts it on the 1249th rank. Conversely, "*long terminal repeat*" is ranked on 242 by t-test, whereas it is ranked on 24 by $P\text{-}Mod$. In fact, even lower-frequency multi-word units gain a prominent ranking if they exhibit limited paradigmatic modifiability. For example, the trigram term "*porphyria cutanea tarda*" is ranked on 28 by $P\text{-}Mod$ although its frequency is only 48 (which results in rank 3291 on the t-test output list). Despite its lower frequency, this term may be judged relevant for the molecular biology domain.[9] It should be noted that the termhood values (and the corresponding list ranks) computed by $P\text{-}Mod$ also include $k = 3$ and hence take into account some frequency factor. As can be seen from the previous ranking examples, however, this factor does not override the paradigmatic modifiability factors of the lower ks.

On the other hand, $P\text{-}Mod$, of course, will also demote true terms in their ranking if their paradigmatic modifiability is less limited. This is particularly the case if one or more of the tokens of a particular term often occur in the same slot of other equal-length n-grams. For example, the trigram term *bone marrow cell* occurs 1757 times in our corpus and is thus ranked quite high (on 31) by t-test. $P\text{-}Mod$, however, ranks this term on 550 because the token *cell* also occurs in many other trigrams and thus leads to a less limited paradigmatic modifiability. Still, the underlying assumption of our approach is that such a case is rather the exception than the rule and that terms are in fact linguistically more fixed than non-terms, which is exactly what our measure of limited paradigmatic modifiability aims at quantifying.

[8] Setting the upper limit of k to n (which would be $n = 3$ for trigrams) actually has the pleasant side effect of including frequency in our modifiability measure. In this case, the only possible selection $k_1k_2k_3$ as the denominator of Formula (2) is equivalent to summing up the frequencies of all trigram term candidates.

[9] It denotes a group of related disorders, all of which arise from deficient activity of the heme synthetic enzyme uroporphyrinogen decarboxylase (URO-D) in the liver.

3.4 Methods of Evaluation

As already described in Subsection 3.2, standard procedures for evaluating the quality of termhood measures usually involve identifying the true positives among an (usually) arbitrarily set number m of the highest ranked candidates returned by a particular measure, which is usually done by a domain expert. Because this is also labor-intensive (besides being unreliable), m is usually small, ranging from 50 to several hundreds.[10] In contrast, by taking a large and already established terminology as an evaluation gold standard, we are able to dynamically examine various m-highest ranked samples, which allows for the plotting of standard precision and recall graphs for the whole candidate set. Through this, we provide a much more reliable evaluation metric for ATR measures than what is typically employed in the literature.

We evaluate our $P\text{-}Mod$ algorithm against the t-test measure,[11] which, of all standard mesures, yields the best results in general-language collocation extraction studies [12], and against the widely used C-value, which aims at enhancing the common frequency of occurrence measure by making it sensitive to nested terms [8]. Our baseline is defined by the proportion of true positives (i.e., the proportion of terms) in our bi-, tri- and quadgram candidate sets, which is equivalent to the likelihood of finding one by blindly picking from one of the different sets (see Subsection 3.2 above).

4 Results and Discussion

4.1 Precision/Recall for Terminology Extraction

For each of the different candidate sets, we incrementally examined portions of the ranked output lists returned by each of the three measures we considered. The precision values for the various portions were computed such that for each percent point of the list, the number of true positives found (i.e., the number of terms found, according to the UMLS METATHESAURUS) was scaled against the overall number of candidate items returned. This yields the (descending) precision curves in Figures 1, 2 and 3 and some associated values in Table 4.

First, we observe that, for the various n-gram candidate sets examined, all measures outperform the baselines by far, and, thus, all are potentially useful measures of termhood. As can be clearly seen, however, our $P\text{-}Mod$ algorithm substantially outperforms all other measures at almost all points for all n-grams examined. Considering 1% of the bigram list (i.e., the first 673 candidates) the precision value for $P\text{-}Mod$ is 20 points higher than for t-test and for C-value. At 1% of the trigram list (i.e., the first 310 candidates), $P\text{-}Mod$'s lead is 7 points. Considering 1% of the quadgrams (i.e., the first 108 candidates), t-test actually leads by 7 points. At 10% of the quadgram list, however, the $P\text{-}Mod$ precision score has overtaken the other ones. With increasing portions of all (bi-, tri-, and quadgram) ranked lists considered, the precision curves start to converge toward the baseline, but $P\text{-}Mod$ maintains a steady advantage.

[10] Studies on collocation extraction (e.g. [12]) also point out the inadequacy of such evaluation methods claiming they usually lead to very superficial judgements about the measures to be examined.

[11] See [11] for a description how this measure can be used for the extraction of multi-word expressions.

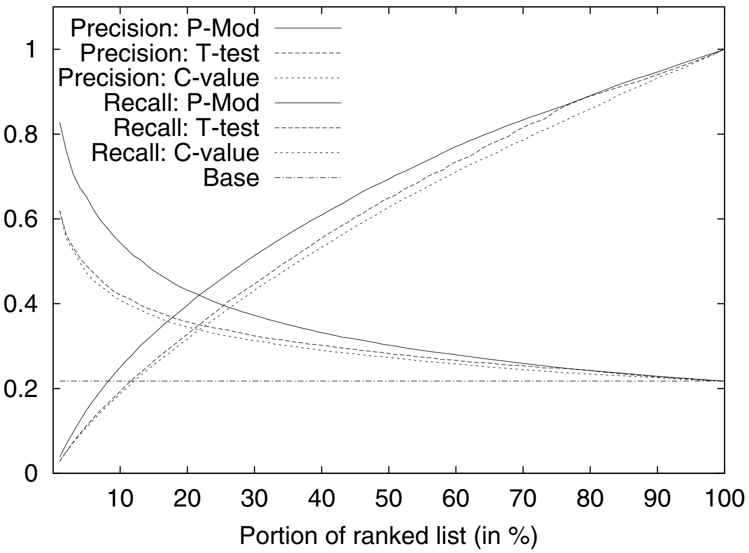


Fig. 1. Precision/Recall for Bigram Biomedical Term Extraction

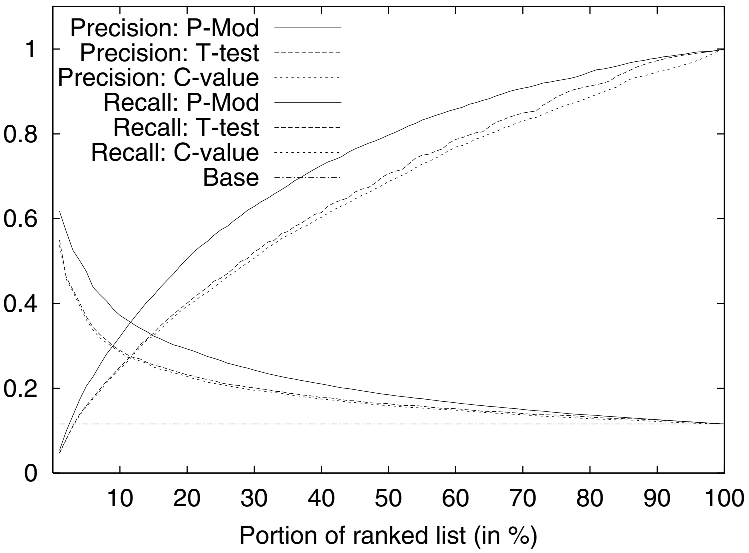


Fig. 2. Precision/Recall for Trigram Biomedical Term Extraction

The (ascending) recall curves in Figures 1, 2 and 3 and their corresponding values in Table 5 indicate which *proportion of all true positives* (i.e., the proportion of all terms in a candidate set, according to the UMLS METATHESAURUS) is identified by a particular measure at a certain point of the ranked list. In this sense, recall is an even better indicator of a particular measure's performance.

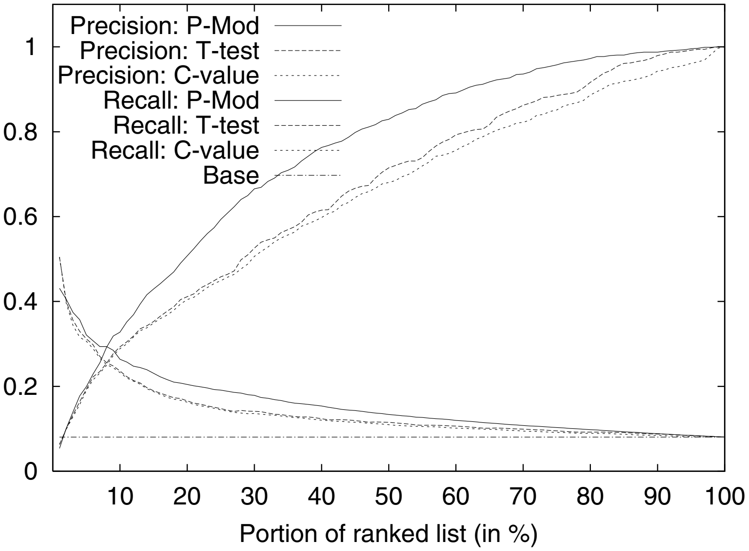


Fig. 3. Precision/Recall for Quadgram Biomedical Term Extraction

Table 4. Precision Scores for Biomedical Term Extraction at Selected Portions of the Ranked Output List

	Portion of ranked list considered	**Precision** scores of measures		
		P-Mod	t-test	C-value
Bigrams	1%	0.82	0.62	0.62
	10%	0.53	0.42	0.41
	20%	0.42	0.35	0.34
	30%	0.37	0.32	0.31
	baseline	0.22	0.22	0.22
Trigrams	1%	0.62	0.55	0.54
	10%	0.37	0.29	0.28
	20%	0.29	0.23	0.23
	30%	0.24	0.20	0.19
	baseline	0.12	0.12	0.12
Quadgrams	1%	0.43	0.50	0.50
	10%	0.26	0.24	0.23
	20%	0.20	0.16	0.16
	30%	0.18	0.14	0.14
	baseline	0.08	0.08	0.08

Again, our linguistically motivated terminology extraction algorithm outperforms all others, and with respect to tri- and quadgrams, its gain is even more pronounced than for precision. In order to get a 0.5 recall for bigram terms, $P\text{-}Mod$ only needs to winnow

Table 5. Portions of the Ranked Output List to consider to obtain Selected Recall Scores for Biomedical Term Extraction

	Recall scores of measures	Portion of Ranked List		
		P-Mod	t-test	C-value
Bigrams	0.5	29%	35%	37%
	0.6	39%	45%	47%
	0.7	51%	56%	59%
	0.8	65%	69%	72%
	0.9	82%	83%	85%
Trigrams	0.5	19%	28%	30%
	0.6	27%	38%	40%
	0.7	36%	50%	53%
	0.8	50%	63%	66%
	0.9	68%	77%	84%
Quadgrams	0.5	20%	28%	30%
	0.6	26%	38%	40%
	0.7	34%	49%	53%
	0.8	45%	62%	65%
	0.9	61%	79%	82%

Table 6. Significance testing of differences for bi-, tri- and quadgrams using the two-tailed McNemar test at 95% confidence interval

# of measure points	# of significant differences comparing *P-Mod* with					
	t-test	C-value	t-test	C-value	t-test	C-value
10	10	10	9	9	3	3
20	20	20	19	19	13	13
30	30	30	29	29	24	24
40	40	40	39	39	33	33
50	50	50	49	49	43	43
60	60	60	59	59	53	53
70	70	70	69	69	63	63
80	75	80	79	79	73	73
90	84	90	89	89	82	83
100	93	100	90	98	82	91
	bigrams		trigrams		quadgrams	

29% of the ranked list, whereas t-test and C-value need to winnow 35% and 37%, respectively. For trigrams and quadgrams, *P-Mod* only needs to examine 19% and 20% of the list, whereas the other two measures need to scan almost 10 additional percentage points. In order to obtain a 0.6, 0.7, 0.8 and 0.9 recall, the differences between the measures narrow for bigram terms, but they widen substantially for tri- and quadgram terms. To obtain a 0.6 recall for trigram terms, *P-Mod* only needs to winnow 27% of its output list while t-test and C-value need to analyze 38% and 40%, respectively. To get 0.7 recall, $P\text{-}Mod$ only needs to analyze 36%, and the second-placed t-test already 50% of the ranked list. For a 0.8 recall, this relation is 50% ($P\text{-}Mod$) to 63% (t-test), and at recall point 0.9, 68% ($P\text{-}Mod$) to 77% (t-test). For quadgram term identification, the results for $P\text{-}Mod$ are equally superior to those for the other measures, and at recall points 0.8 and 0.9 even more pronounced than for trigram terms.

We also tested the significance of differences for our results, both between $P\text{-}Mod$ and t-test and between $P\text{-}Mod$ and C-value. Because in all cases the ranked lists were taken from the same set of candidates (*viz.* the set of bigram candidate types, the set of trigram candidate types, or the set of quadgram candidate types), and hence constitute dependent samples, we applied the McNemar test [19] for statistical testing. We selected 100 measure points in the ranked lists, one after each increment of one percent, and then used the two-tailed test for a confidence interval of 95%. Table 6 lists the number of significant differences for these measure points at intervals of 10 for the bi-, tri-, and quadgram results. For the bigram differences between $P\text{-}Mod$ and C-value, all of them are significant, and between $P\text{-}Mod$ and t-test, all are significantly different up to measure point 70.[12] Looking at the tri- and quadgrams, although the number of significant differences is less than for bigrams, the vast majority of measure points still is significantly different and thus underlines the superiour performance of the $P\text{-}Mod$ measure.

5 Conclusions

In our study, we proposed a new terminology identification algorithm and showed that it substantially outperforms some of the standard measures in distinguishing terms from non-terms in the biomedical literature. While mining technical and scientific literature for new terminological units and assembling those in controlled vocabularies is an overall complex task involving several components, one essential building block is a measure indicating the *degree of termhood* of a candidate. In this respect, our study has shown that an algorithm which incorporates a vital linguistic property of terms, *viz.* their *limited paradigmatic modifiability*, can be a much more powerful and valuable part of a terminology extraction system (like, e.g., proposed by [20]) than the standard measures that are typically employed.

In general, a high-performing term identification system is not only valuable for collecting new terms per se but is also essential in updating already existing terminology resources. As a concrete example, the term "*cell cycle*" is contained in the hierarchically-structured biomedical MESH terminology and the term "*cell cycle arrest protein BUB2*" in the MESH supplementary concept records which include many proteins with a GENBANK[21][13] identifier. The word trigram *cell cycle arrest*, however, is not included in MESH although it is ranked in the top 10% of $P\text{-}Mod$. Utilizing this prominent ranking, the missing semantic link can be established between these two

[12] As can be seen in Figures 1, 2 and 3 above, the curves start to merge at the higher measure points and thus the number of significant differences decreases.

[13] GENBANK is a database containing an annotated collection of all publicly available DNA sequences.

terms (i.e., between *cell cycle* and *cell cycle arrest protein BUB2*), both by including the trigram *cell cycle arrest* in the MESH hierarchy and by linking it via the comprehensive terminological umbrella system for biomedicine, *viz.* UMLS, to the Gene Ontology (GO [22]), in which it is listed as a stand-alone term.

References

[1] Nakagawa, H., Mori, T.: Nested collocation and compound noun for term recognition. In: COMPUTERM '98 – Proceedings of the First Workshop on Comutational Terminology. (1998) 64–70

[2] Hersh, W.R., Campbell, E., Evans, D., Brownlow, N.: Empirical, automated vocabulary discovery using large text corpora and advanced natural language processing tools. In Cimino, J.J., ed.: AMIA'96 – Proceedings of the 1996 AMIA Annual Fall Symposium (formerly SCAMC). Beyond the Superhighway: Exploiting the Internet with Medical Informatics, Washington, D.C., October 26-30, 1996. Philadelphia, PA: Hanley & Belfus (1996) 159–163

[3] Rindflesch, T.C., Hunter, L., Aronson, A.R.: Mining molecular binding terminology from biomedical text. In Lorenzi, N.M., ed.: AMIA'99 – Proceedings of the 1999 Annual Symposium of the American Medical Informatics Association. Transforming Health Care through Informatics: Cornerstones for a New Information Management Paradigm, Washington, D.C., November 6-10, 1999. Philadelphia, PA: Hanley & Belfus (1999) 127–131

[4] Collier, N., Nobata, C., Tsujii, J.: Automatic acquisition and classification of terminology using a tagged corpus in the molecular biology domain. Terminology **7** (2002) 239–257

[5] Bodenreider, O., Rindflesch, T.C., Burgun, A.: Unsupervised, corpus-based method for extending a biomedical terminology. In: Proceedings of the ACL Workshop on Natural Language Processing in the Biomedical Domain., Pittsburgh, PA, USA. Association for Computational Linguistics (2002) 53–60

[6] Nenadić, G., Spasic, I., Ananiadou, S.: Terminology-driven mining of biomedical literature. Journal of Biomedical Informatics **33** (2003) 1–6

[7] Krauthammer, M., Nenadić, G.: Term identification in the biomedical literature. Journal of Biomedical Informatics **37** (2004) 512–526

[8] Frantzi, K., Ananiadou, S., Mima, H.: Automatic recognition of multi-word-terms: the C/NC value method. International Journal of Digital Libraries **3** (2000) 115–130

[9] Nenadić, G., Ananiadou, S., McNaught, J.: Enhancing automatic term recognition through recognition of variation. In: COLING 2004 – Proceedings of the 20th International Conference on Computational Linguistics, Association for Computational Linguistics (2004) 604–610

[10] Damerau, F.J.: Generating and evaluating domain-oriented multi-word terms from text. Information Processing & Management **29** (1993) 433–447

[11] Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. Cambridge, MA; London, U.K.: Bradford Book & MIT Press (1999)

[12] Evert, S., Krenn, B.: Methods for the qualitative evaluation of lexical association measures. In: ACL'01 – Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, Toulouse, France (2001) 188–195

[13] Wermter, J., Hahn, U.: Collocation extraction based on modifiability statistics. In: COLING Geneva 2004 – Proceedings of the 20th International Conference on Computational Linguistics. Volume 2., Geneva, Switzerland, August 23-27, 2004. Association for Computational Linguistics (2004) 980–986

[14] Kudo, T., Matsumoto, Y.: Chunking with support vector machines. In: NAACL'01, Language Technologies 2001 – Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics, Pittsburgh, PA, USA, June 2-7, 2001 (2001) 192–199

[15] Justeson, J.S., Katz, S.M.: Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering **1** (1995) 9–27

[16] Browne, A.C., Divita, G., Nguyen, V., Cheng, V.C.: Modular text processing system based on the SPECIALIST lexicon and lexical tools. In Chute, C.G., ed.: AMIA'98 – Proceedings of the 1998 AMIA Annual Fall Symposium. A Paradigm Shift in Health Care Information Systems: Clinical Infrastructures for the 21st Century, Orlando, FL, November 7-11, 1998. Philadelphia, PA: Hanley & Belfus (1998) 982

[17] UMLS: Unified Medical Language System. Bethesda, MD: National Library of Medicine (2004)

[18] MESH: Medical Subject Headings. Bethesda, MD: National Library of Medicine (2004)

[19] Sachs, L.: Applied Statistics: A Handbook of Techniques. 2nd edn. New York: Springer (1984)

[20] Mima, H., Ananiadou, S., Nenadić, G.: The ATRACT workbench: Automatic term recognition and clustering of terms. In Matusek, V., ed.: Text, Speech and Dialog (TSD 2001). Volume 2166 of Lecture Notes in Artificial Intelligence. Berlin: Springer (2001) 126–133

[21] Benson, D.A., Boguski, M.S., Lipman, D.J., Ostell, J., Ouellette, F.B., Rapp, B.A., Wheeler, D.L.: GENBANK. Nucleic Acids Research **27** (1999) 12–17

[22] Gene Ontology Consortium: Creating the Gene Ontology resource: Design and implementation. Genome Research **11** (2001) 1425–1433

Active Constrained Clustering by Examining Spectral Eigenvectors

Qianjun Xu[1], Marie desJardins[1], and Kiri L. Wagstaff[2]

[1] University of Maryland Baltimore County, Dept. of CS&EE,
1000 Hilltop Circle, Baltimore MD 21250
{qxu1, mariedj}@cs.umbc.edu

[2] Jet Propulsion Laboratory, California Institute of Technology,
4800 Oak Grove Dr., Pasadena CA 91109
kiri.wagstaff@jpl.nasa.gov

Abstract. This work focuses on the active selection of pairwise constraints for spectral clustering. We develop and analyze a technique for Active Constrained Clustering by Examining Spectral eigenvectorS (ACCESS) derived from a similarity matrix. The ACCESS method uses an analysis based on the theoretical properties of spectral decomposition to identify data items that are likely to be located on the boundaries of clusters, and for which providing constraints can resolve ambiguity in the cluster descriptions. Empirical results on three synthetic and five real data sets show that ACCESS significantly outperforms constrained spectral clustering using randomly selected constraints.

1 Introduction

Recently, clustering research has focused on developing methods to incorporate domain knowledge into clustering algorithms, so that the results are tailored to the interests and existing knowledge of the user. For example, pairwise constraints were introduced by Wagstaff *et al.* [1] as a way to use domain-specific information in the form of *must-link* constraints, which specify that two instances must be in the same cluster, and *cannot-link* constraints, which indicate that two instances must be in different clusters. Although it has been repeatedly demonstrated that constraints can improve clustering performance [1,2,3,4], these gains often require the user to specify constraints for a significant fraction of the items in the data set. In this paper, we seek to reduce that user burden by *actively* selecting item pairs for constraint labelling, so that the most informative constraints are acquired as quickly as possible.

Active constraint selection has been previously studied by Basu *et al.* for the K-means algorithm [5]. Their method aims to find k neighborhoods to initialize the clusters. However, for data sets that have close boundaries or small overlap areas on the boundaries, which are the focus of this paper, this method does not work well. We instead propose an active constraint selection method that identifies crucial *boundary points* (those near cluster boundaries) with high probability.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 294–307, 2005.

The main contribution of this paper is an active constraint selection technique for data sets with close or overlapping boundaries. We refer to this method as Active Constrained Clustering by Examining Spectral eigenvectorS (ACCESS). ACCESS uses a heuristic derived from the theoretical properties of spectral decomposition methods to identify points at or near cluster boundaries with high probability. Providing the clustering algorithm with constraints on such points can help to resolve ambiguity in the cluster descriptions. Our experiments on three synthetic and five real data sets show that ACCESS yields a significant performance improvement over constrained clustering with randomly selected constraints.

2 Background

Spectral Clustering. The eigenvectors derived from the data similarity graph have good properties and can be used for clustering; this class of methods is referred to as *spectral clustering* techniques. Given n data points, we can construct a graph $G = (V, E, A)$, where each vertex v_i corresponds to a point p_i, and the edge $e_{i,j}$ between vertices i and j is weighted by their (dis)similarity value, $a_{i,j}$. Any similarity measure can be used; one popular similarity metric is defined as

$$A_{i,j} = exp(\frac{-\delta_{ij}^2}{2\sigma^2}), \tag{1}$$

where δ_{ij} is the Euclidean distance between point i and j and σ is a free scale parameter. Using this definition, the larger the distance δ_{ij}, the smaller the similarity A_{ij}.

The goal of spectral clustering is to find a *minimal cut* of the graph such that the inter-cluster similarities are minimized. However, this objective favors cutting off a small number of isolated points [6]. Previous research explored refined objectives to overcome this drawback, including the *ratio cut* [6] and *normalized cut* [7] criteria. It can be shown that the second smallest eigenvector of the (generalized) graph Laplacian matrix, defined as $L = D - A$, where D is a diagonal matrix with element $d_{ii} = \sum_{j=1}^{n} A_{ij}$, is an approximation of the cluster membership indicator vector and its corresponding eigenvalue gives the optimal cut value [6,7]. The second smallest eigenvector is used to split the data into two groups.

Recently, researchers have proposed to make use of k eigenvectors simultaneously for the multi-cluster problem [8,9]. These methods usually use a normalized similarity graph, such as

$$P = D^{-1}A \tag{2}$$

or

$$N = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}. \tag{3}$$

Note that the eigenvectors derived from the P and N matrices are related to the eigenvectors derived from the (generalized) Laplacian matrix. In particular, if λ and x are the solutions of Equation 2, then $1 - \lambda$ and x are the solutions of the generalized Laplacian matrix [9]. After obtaining k eigenvectors, any clustering technique, such as the K-means algorithm, can be applied to the eigenspace, that is, the space spanned by the largest k eigenvectors. The justification of clustering in the eigenspace can be found in Ng *et al.* [8] and Meila *et al.* [9].

Each row in the P matrix sums to 1. Therefore, we can interpret the entries P_{ij} as the transition probabilities of a random walker moving from point i to point j. The probabilistic interpretation of the normalized similarity matrix gives an intuitive explanation of the constraints, as we will discuss next.

Incorporating Constraints in Spectral Clustering. Kamvar *et al.* developed a technique to incorporate constraints into spectral clustering [10]. We will refer to their method as KKM after the authors' initials. Their work uses a different normalization matrix, as follows:

$$N = (A + d_{max}I - D)/d_{max}, \tag{4}$$

where d_{max} is the largest element in the matrix D and I is the identity matrix. Note that the off-diagonal entries of N are simply the scaled similarity values: $N_{ij} = A_{ij}/d_{max}$ for $i \neq j$. The diagonal entries, however, are computed by $N_{ii} = (d_{max} - d_{ii})/d_{max}$.

Given a must-link constraint (i, j), KKM modifies the corresponding affinities so that $A_{ij} = A_{ji} = 1$; as a result, when N is re-derived from the new similarity matrix, the transition probability between i and j will be greater than or equal to the transition probabilities leading from i or j to any other point. Similarly, a cannot-link constraint (i, j) is incorporated by setting $A_{ij} = A_{ji} = 0$, preventing a direct transition between points i and j.

Note that the use of the transition probability matrix in Equation 4 may cause problems when there are outliers in the data. For example, if point i is isolated from all other data points, then N_{ii} will be much larger than all other entries N_{ij}. Therefore, once a random walk encounters point i, it has a very low probability of leaving it, resulting in a singleton cluster. To overcome this drawback, our method replaces KKM's transition matrix N with the P matrix in Equation 2. We discuss the advantages of using this matrix in Section 5.

Notation. In this paper, we focus on the two-cluster problem, and assume that there are only two clusters, C_1 and C_2. We index the points so that the points in the first cluster appear before the points in the second cluster. We write the similarity matrix $A = (A_{C_1C_1}, A_{C_1C_2}; A_{C_2C_1}, A_{C_2C_2})$, where $A_{C_1C_1}$ and $A_{C_2C_2}$ are the intra-cluster similarity sub-matrices, and $A_{C_1C_2} = A^T_{C_2C_1}$ are the inter-cluster similarity sub-matrices.

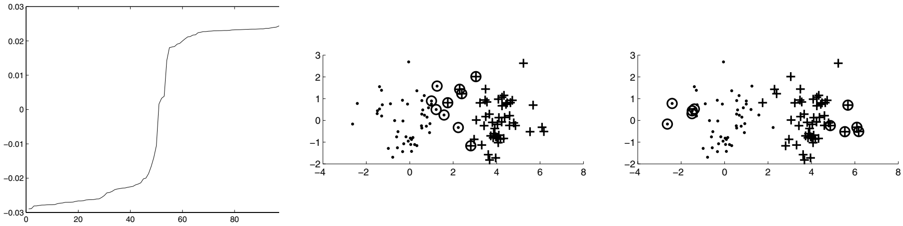
3 Active Constraint Selection

We are interested in clustering problems where the clusters are nearly separated – by which we mean that the boundaries of the clusters are very close, and there may be small overlapping areas. We propose to first analyze the eigenvectors derived from the data similarity matrix to identify sparse points and boundary points. Then we will query an oracle to give us correct pairwise constraints on these ambiguous points. Incorporating these constraints into spectral clustering improves performance.

Properties of Eigenvectors. Our active constraint selection method is based on the following properties of the eigenvectors. Interested readers are referred to the citations for proofs of the theoretical results that we use here.

1. In the ideal case, if the k clusters are well separated, so that only the intra-cluster similarity sub-matrices have nonzero entries, we will obtain k *piecewise constant* eigenvectors—in other words, items from the same cluster will have the same values in the eigenvector, and the clusters can be easily recognized [11,12].
2. If the clusters are nearly separated (i.e., the dense clusters are loosely connected by a few *bridges* (edges) between them), then the first k eigenvectors will be approximately piecewise constant. This claim has previously been shown by applying matrix perturbation theory to the ideal case [13]. The values in the eigenvectors of points adjacent to these bridges will be pulled towards each other.
3. If the graph is connected, then the identity vector $\mathbf{1}$ is the smallest eigenvector of the Laplacian matrix, and the corresponding eigenvalue is 0. All other eigenvectors are orthogonal to $\mathbf{1}$, which implies that there are both positive and negative (and possibly zero) values in each eigenvector. This can be easily shown by the definition of the Laplacian matrix. This fact motivates a simple heuristic to partition the data: items with positive values in the eigenvector can be put into one cluster, and items with negative values in the eigenvector can be put into the other cluster [11].
4. It has been proved [6] that the second smallest eigenvector of the Laplacian matrix gives the optimal ratio cut cost for splitting the data set into two groups. By inference, the third smallest eigenvector gives the optimal ratio cut cost for further splitting the first two groups. A similar result has been derived for the generalized eigenvectors of the Laplacian matrix for the normalized cut criterion[7]. In summary, the sorted eigenvalues indicate the estimate of cut cost in order, and the different eigenvectors correspond to different splitting strategies.

Close and Distant Boundary Points. For the scenario we are interested in, the items located on the cluster boundaries are the objectives of our active constraint selection, since they are far from the cluster centers and may be interspersed with boundary points of the other clusters. If we can impose constraints to strengthen the similarity between boundary points and members of their clusters, while weakening their similarity to points from other clusters, the clusters themselves will be more clearly apparent in the similarity matrix. We distinguish boundaries between clusters from the outer boundaries of clusters by calling the former *close boundaries*, and the latter *distant boundaries*. Our method aims to find both types.



(a)The 2nd largest eigenvector (b)Close boundary points (c)Distant boundary points

Fig. 1. An illustration of the active constraint selection

According to statement 2 above, for the data sets we are interested in, the eigenvector will be approximately piecewise constant. In this case, the values in the eigenvector will be bimodally distributed and centered on v and $-w$ (where v and w are positive numbers), with some variances. The close boundary points, *i.e.*, items adjacent to the bridges, are likely to have eigenvector values towards the opposite center. In addition, we hypothesize that the items with eigenvector values far from 0 will be on the distant boundaries (see Figure 1).

Figure 1(a) shows the sorted second largest eigenvector of two Gaussian distributed clusters (represented by + and ·) and the close and distant boundary points identified from this eigenvector (represented by o). The 10 points with eigenvector values closest to 0 are shown in Figure 1(b); they indeed appear to be located on the close boundaries. In Figure 1(c), the points with largest positive (negative) values have been identified as distant boundary points.

Since we only consider clustering problems with two clusters, we expect that the largest two eigenvectors of the P matrix will be most useful for splitting the data. However, whether or not these eigenvectors are appropriate for this purpose depends on the true data distribution and on the value of the σ parameter, as we show next.

Sparse Points. Figure 2 shows the Ellipses data set. It has two important characteristics: (1) the two clusters have very close boundaries and there is a small group of overlapped items; and (2) there are two small groups of '+' data—the three circled items located at the bottom left corner and one located at the top right corner in Figure 2—that are far away from the main group of '+' data. We call these *sparse points*, and we now examine their effects on the eigenvectors.

The distance from these sparse points to the center of the cluster to which they belong is larger than the distance between the boundaries of two clusters. Therefore, for small values of σ, it is possible that the largest eigenvectors will treat these small groups of '+' data items as a separate cluster. This is exactly what we see using the similarity matrix with $\sigma = 0.2$. Figure 3(a) shows the second largest eigenvector of the Ellipses data set. The anomalous points are exactly those sparse points in Figure 2. Fortunately, the third largest eigenvector (Figure 3(b)) roughly corresponds to the groupings for the remaining data. From Figure 3(b) we can see that most of the data in the first cluster (indexed from 1

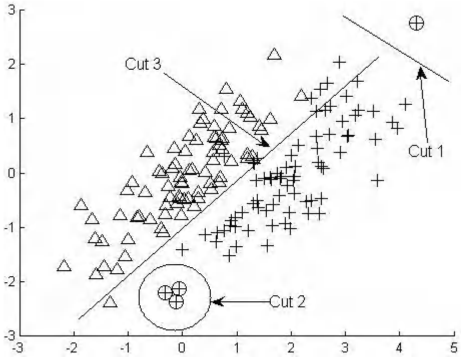
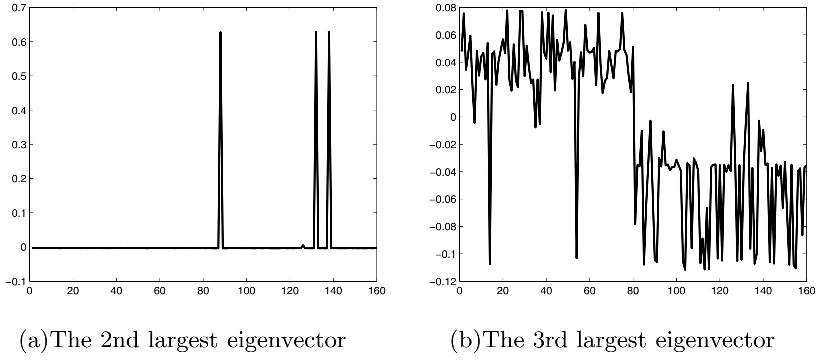


Fig. 2. Ellipses data set

to 80) have positive values, while most data in the second cluster have negative values. Several items violate this structure. These items have either values near zero or large negative values, and therefore can be identified by our method. We can interpret the eigenvectors as follows: the first eigenvector gives cut 1 in Figure 2, the second eigenvector gives cut 2, and the third eigenvector gives cut 3. The third largest eigenvector will be automatically selected by ACCESS to identify the close and distant boundary points.



(a)The 2nd largest eigenvector (b)The 3rd largest eigenvector

Fig. 3. The eigenvectors of Ellipses data set

In summary, for two-cluster problems, our active constraint selection method will identify two types of informative points: (1) the sparse points, identified by the first m eigenvectors (where m depends on how many sparse subclusters are found in the data set), and (2) the close and distant boundary points identified by the $(m+1)$st eigenvector. These $m+1$ eigenvectors are used to construct the eigenspace matrix in step 7 of the ACCESS algorithm, given below. In the next section, we explain how we identify these points.

```
Derive matrix $A$ and matrix $P = D^{-1}A$.
Compute the eigenvalues and eigenvectors of $P$.
Actively pick $q$ data points by examining the eigenvectors and query the oracle for
  labels or pairwise constraints.
Impose must-link constraint pairs $(i, j)$ by assigning $A_{ij} = A_{ji} = 1$.
Impose cannot-link constraint pairs $(i, j)$ by assigning $A_{ij} = A_{ji} = 0$.
Reconstruct matrix $P'$.
Identify the largest $m'$ eigenvectors that have sparse points.
Pick the largest $m' + 1$ largest eigenvectors of $P'$, and construct the eigenspace
  matrix $X = (x^1, x^2, \cdots, x^{m'+1})$.
Row normalize $X$ to length 1.
Perform K-means clustering on the rows of $X$ to identify two clusters.
Assign data point $i$ to cluster $c$ if row $X_{i.}$ is assigned to cluster $c$.
```

Fig. 4. The ACCESS algorithm

Implementation of Active Constraint Selection. Active constraint selection starts with the largest eigenvector. Each eigenvector may produce one of two outcomes. If it identifies one or more sparse points (defined as points whose deviation from the mean value in the eigenvector are greater than three standard deviations), then the next eigenvector will be further examined. Alternatively, if it does not identify any sparse points, then we use this $(m+1)$st eigenvector to identify the close and distant boundary points, and we ignore the remaining eigenvectors.

The close and distant boundary points are identified as follows. Each data point has an associated p_{close}-*value* and $p_{distant}$-*value* when considered as a close or distant boundary point, respectively. The p_{close}-value is inversely proportional to its distance from 0, while the $p_{distant}$-value is proportional to its distance from 0. The detailed computation is as follows (ϵ is a small constant):

```
for the $(m+1)st$ eigenvector $e$
$max_{pos} = \max\{e_i \mid e_i >= 0\}$
$max_{neg} = \max\{abs(e_i) \mid e_i < 0\}$
for $e_i$ do
  if $e_i >= 0$ then
    $p^i_{close} = (max_{pos} - e_i + \epsilon)/(max_{pos} + \epsilon)$
    $p^i_{distant} = (e_i + \epsilon)/(max_{pos} + \epsilon)$
  else
    $p^i_{close} = (max_{neg} - abs(e_i) + \epsilon/(max_{neg} + \epsilon))$
    $p^i_{distant} = (abs(e_i) + \epsilon)/(max_{neg} + \epsilon)$
  end if
end for
```

Our method chooses sets of boundary points S_{close} and $S_{distant}$ such that following condition is satisfied: $\{p^i_{close} >= p^j_{close}, \forall i, j, i \in S_{close}, j \notin S_{close}\}$ and $\{p^i_{distant} >= p^j_{distant}, \forall i, j, i \in S_{distant}, j \notin S_{distant}\}$. Given q, the number of points to query, ACCESS selects s sparse points, $2(q-s)/3$ close boundary points and $(q-s)/3$ distant boundary points.

Algorithm. The pseudo-code for the ACCESS algorithm is given in Figure 4. There are two parameters: q, the number of items to query, and σ, the scale parameter in Equation 1. Note that our main contribution is in step 3, active constraint selection.

4 Experiments and Results

Data Sets. We implemented experiments on three synthetic and five real data sets. The Sphere data set is generated by Gaussian distributions with mean $(0, 0)$ and $(3, 0)$, and covariance matrix $(1, 0; 0, 1)$. The Ellipses and Test data sets are shown in Figure 2 and Figure 5. The Iris and Soybean data sets are from the UCI Machine Learning Repository [14]. For these data sets, we derive the similarity matrix from the Euclidean distances as in Equation 1. The text data sets are from the 20 Newsgroups collection. We preprocess the data as described by Basu *et al.* [5], then use cosine similarity values. Let $NN_{20}(p)$ be the set of 20 nearest neighbors to point p. We set $A_{i,j}$ of the similarity matrix to zero if $p_i \notin NN_{20}(j)$ and $p_j \notin NN_{20}(i)$. The value 20 was selected based on the method reported by Kamvar *et al.* [10]. Key properties of each data set are shown in Table 1.

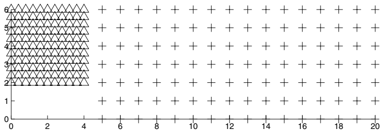


Fig. 5. Tester data set

Table 1. Real data sets

data	cluster 1	cluster 2	num
Iris	Versicolour	Virginica	100
Soybean	brown-spot	frog-eye-leaf-spot	183
Text1	alt.atheism	rec.sport.baseball	200
Text2	alt.atheism	sci.space	200
Text3	rec.sport.baseball	sci.space	200

Parameter Selection. The σ parameter in Equation 1 significantly affects clustering performance. Ng *et al.* [8] proposed a parameter selection criterion based on the observation that a good σ parameter will yield a partition with small distortion (*i.e.*, small mean squared error). In our implementation, we use a small σ value, since this yields a sparse similarity matrix, which tends to produce good spectral clustering results. In addition, we automatically identify eigenvectors that will isolate small groups of data (Figure 3(b)) and use $m + 1$ eigenvectors for clustering.

Evaluation. The Rand index [15] is often used as an evaluation of the clustering result. The Rand index measures the agreement of two partitions, P_1 and P_2. Given a data set with n points, there are $n(n-1)/2$ pairs of decisions: for each pair of items, each partition either assigns them to the same cluster or to different clusters. Let a and b be the number of pairs for which the two partitions agree by assigning them to the same cluster or to different clusters, respectively. The Rand index (RI) is then defined as:

$$RI(P_1, P_2) = \frac{a+b}{n(n-1)/2}. \tag{5}$$

In other words, the RI computes the percentage of agreements among all pairs of decisions.

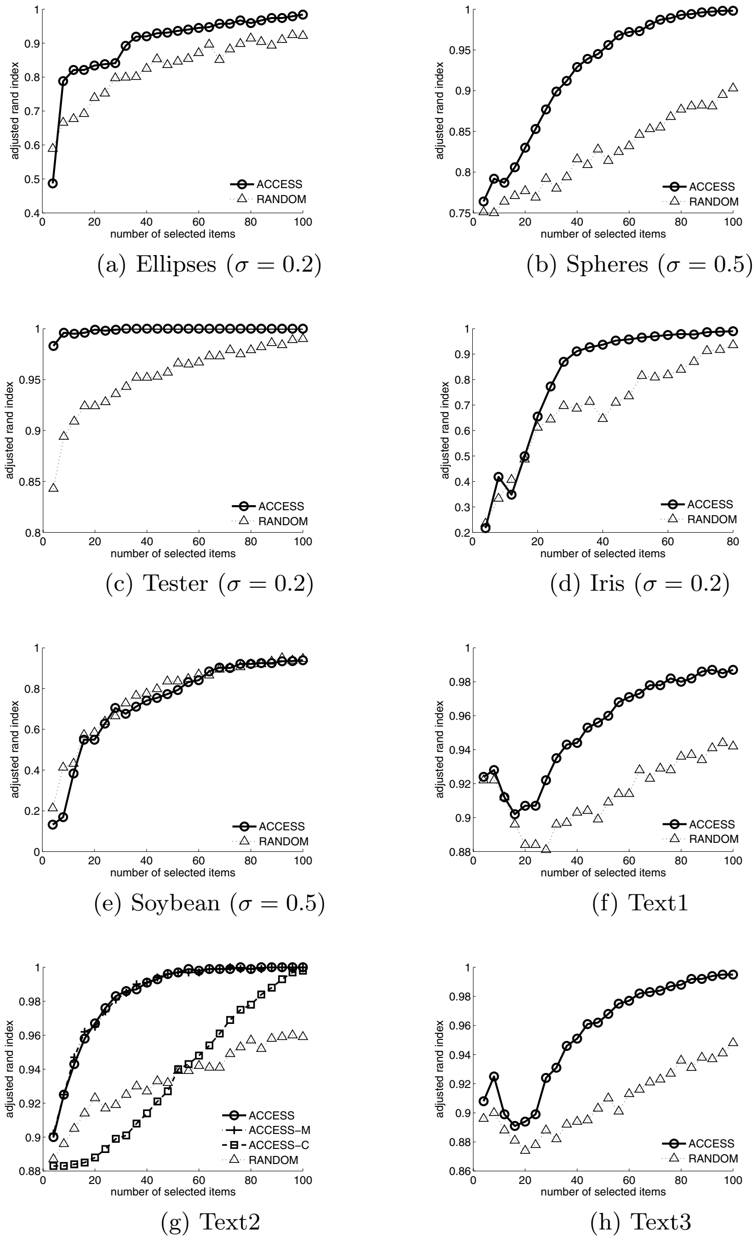
One problem with the Rand index is that its expected value for two random partitions is not a constant. The adjusted Rand index (ARI) [16] has been proposed to overcome this shortcoming. The expected value for two random partitions with a fixed number of clusters for each partition and a fixed number of instances for each cluster is zero. Let n_{ij} be the number of items that appear in cluster i in P_1 and in cluster j in P_2. ARI is computed as:

$$ARI(P_1, P_2) = \frac{R - E[R]}{M[R] - E[R]}, \ R = \sum_{ij} \binom{n_{ij}}{2} \tag{6}$$

where $E[R] = \left[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2}\right] / \binom{n}{2}$ is the expected value of R and $M[R] = \frac{1}{2}\left[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}\right]$ is the maximum possible value for R. Note that, the ARI is usually smaller than the RI. We use the ARI for evaluating the expect of our clustering result with the a priori assigned class labels.

Results and Analysis. The baseline for our experiments is constrained clustering with randomly selected constraints. The items are randomly selected, and constraints for each pair of selected items are derived from their true class labels. We compute the transitive closure of the must-link and cannot-link constraints as in Wagstaff *et al.* [1]. Results are averaged over 100 runs. In the results shown in Figures 6(a) to 6(h), the x axis is the number of items selected, and the y axis is the adjusted Rand index. Note that the only difference between the baseline and our method is which items are selected for querying.

ACCESS yields better performance, with fewer queries, than randomly selecting constraints, on all data sets except Soybean (for which the performance of ACCESS and random selection is approximately equal). To further understand why our method selects good constraints, we examine the similarity matrix for the Text2 data set, before (Figure 7 (a)) and after (Figure 7 (b)) imposing constraints derived from 50 actively selected items (636 must-link and 589 cannot-link constraints). Rows and columns correspond to the item indices. A dot at position (i, j) means that the similarity value $A_{i,j}$ is positive. We first obtain



(a) Ellipses ($\sigma = 0.2$)

(b) Spheres ($\sigma = 0.5$)

(c) Tester ($\sigma = 0.2$)

(d) Iris ($\sigma = 0.2$)

(e) Soybean ($\sigma = 0.5$)

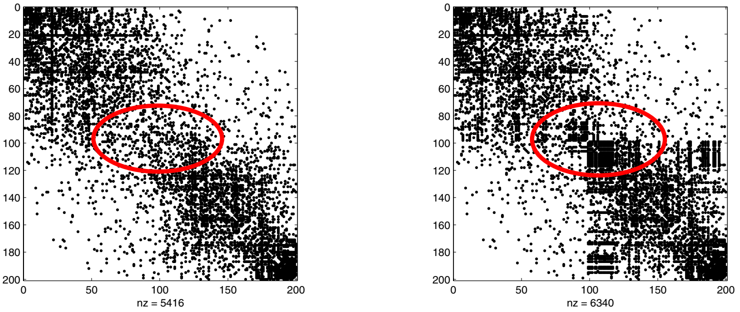
(f) Text1

(g) Text2

(h) Text3

Fig. 6. Performance plots

the second largest eigenvector of the similarity matrix (before imposing any constraints), and then sort the matrix according to the ascending order of this eigenvector. Both plots have the same item ordering. From Figure 7 (a), we can see that the items at the cluster boundaries (i.e., at the intersection of the two



(a) Before

(b) After

Fig. 7. The similarity matrix of the Text2 data set before and after imposing constraints

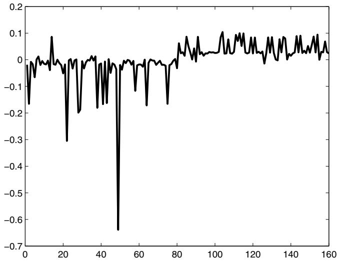


Fig. 8. The sixth largest eigenvector of the Ellipses data set derived using the N matrix in Equation 4

diagonal blocks) are mixed together. After imposing the constraints, they are more clearly distinguished (Figure 7 (b)).

We also did comparative experiments when imposing only must-link or only cannot-link constraints. The results show that the must-link constraints improve clustering performance more than the cannot-link constraints. For some of the data sets (Figure 6(g)), imposing only must-link constraints achieves the same performance as imposing both types of constraints. In Figure 6(g), the curve labeled 'ACCESS-M' shows the result of imposing only actively selected must-link constraints, while the 'ACCESS-C' curve illustrates the result of imposing only actively selected cannot-link constraints.

Figure 6(e) shows a case where our method is less effective. Further examining the Soybean data set, there are large overlapping areas between the two clusters. In this case, our method performs comparably to randomly selected constraints.

5 Discussion and Related Work

There are two primary reasons that we used the P matrix given in Equation 2, rather than the N matrix used by KKM in Equation 4. First, we have previously showed that using randomly selected constraints, the KKM method sometimes performs worse (and was not seen to perform better) than the P-based method [17]. Second, and more importantly, due to the disadvantages discussed in Section 2, the eigenvectors derived from the N matrix fail to identify close and distant boundary points. Because of the complicated distribution of clusters, the N matrix often yields eigenvectors in which several sparse points have extremely large values, while all other points have values near 0. For example, for the N matrix of the Ellipses data set, ACCESS identifies sparse points in the first five eigenvectors. The sixth eigenvector is plotted in Figure 4. However, even for this eigenvector, our method for identifying sparse points returns new points. As a result, it is difficult to use the N matrix to identify the boundary points in the data set.

Recently, there has been some work on active constrained clustering in general. Basu *et al.* implemented an active constraint selection for their Pairwise Constrained K-means algorithm [5]. Their method has two phases. The first phase, Explore, selects an k-neighborhood of must-linked points using the k-centers heuristic. This k-neighborhood is used to initialize the cluster centroids. When queries are allowed, the Consolidate phase is invoked to randomly select a point and query the user about its relation to the known neighborhoods until a must-link is obtained. The authors proved that at least one point can be obtained for each cluster in at most $k\binom{k}{2}$ queries. It implies that 2-neighborhoods can be obtained after querying four items using their method. After that, the authors suggest invoking the Consolidate phase as early as possible, to randomly select items for querying, because the randomly selected samples capture the underlying data distribution and can produce a better estimate of centroids. Their method is tailored to the K-means algorithm, and the purpose of active selection is to get a good estimate of the cluster centroids. When applying their method to spectral clustering with a large number of selected items (>4 items for 2-cluster problems), the performance should be similar to that of randomly selected items because of the Consolidate phase. An empirical comparison of ACCESS and PCK-means is described in another paper [18].

Klein *et al.* developed a cluster-level active querying technique for hierarchical clustering, which works on data sets that exhibit local proximity structure – locally a point has the same cluster membership as its closest neighbors, while globally, a subcluster has different cluster memberships from its closest neighboring subclusters [2]. These active techniques do not work well in our scenario, where the boundaries are very close. In contrast, our method can identify the points close to the boundaries of clusters.

6 Conclusions and Future Work

In this paper, we described ACCESS, an active constrained spectral clustering method. The actively selected constraints significantly improve clustering performance over randomly selected constraints for data sets that have close boundaries and overlapping regions.

The constraints selected by our method are located on the boundaries of the clusters. It is likely that they could also improve the performance of other clustering methods such as K-means and hierarchical clustering. We are working on applying these constraints to these clustering methods and comparing the performances of different active selection methods.

Our current method focuses on two-cluster problems. We believe that the same idea can be generalized to multiple-cluster problems as well, by identifying the boundary points of one cluster and splitting these points, then recursively splitting the remaining data items.

Acknowledgements

We thank Matthew Gaston, Eric Eaton, Blazej Bulka and Priyang Rathod for helpful discussions on developing this technology. We also thank the anonymous reviewers for their comments on improving this paper. This research is supported by NSF grant 0325329.

References

1. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: Proceedings of the 18th International Conference of Machine Learning. (2001) 577–584
2. 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 of the Nineteenth International Conference on Machine Learning. (2002) 307–314
3. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relations. In: Proceedings of the Twentieth International Conference on Machine Learning. (2003) 11–18
4. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: Proceedings of the Twenty-First International Conference on Machine Learning. (2004) 81–88
5. Basu, S., Banerjee, A., Mooney, R.J.: Active semi-supervision for pairwise constrained clustering. In: Proceedings of the SIAM International Conference on Data Mining. (2004) 333–344
6. Hagen, L., Kahng, A.B.: New spectral methods for ratio cut partitioning and clustering. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **11** (1992) 1074–1085
7. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. (1997) 731–737
8. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14. (2002) 849–856
9. Meilă, M., Shi, J.: A random walks view of spectral segmentation. In: Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics. (2001)
10. Kamvar, S.D., Klein, D., Manning, C.D.: Spectral learning. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence. (2003) 561–566

11. Fiedler, M.: A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. Czechoslovak Mathematical Journal **25** (1975) 619–627
12. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2001) 269–274
13. Ding, C.H., He, X., Zha, H.: A spectral method to separate disconnected and nearly-disconnected web graph components. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2001)
14. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)
15. Rand, W.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association **66** (1971) 846–850
16. Hubert, Arabie: Comparing partitions. Journal of Classification **2** (1985) 193–218
17. Xu, Q., desJardins, M., Wagstaff, K.L.: Constrained spectral clustering under a local proximity structure assumption. In: Proceedings of the 18th International FLAIRS Conference. (2005)
18. Xu, Q., desJardins, M., Wagstaff, K.L.: Active constraint selection for clustering. (Submitted to ICDM 2005)

Learning Ontology-Aware Classifiers

Jun Zhang, Doina Caragea, and Vasant Honavar

Artificial Intelligence Research Laboratory,
Department of Computer Science,
Iowa State University,
Ames, Iowa 50011-1040, USA
{jzhang, dcaragea, honavar}@cs.iastate.edu

Abstract. Many practical applications of machine learning in data-driven scientific discovery commonly call for the exploration of data from multiple points of view that correspond to explicitly specified ontologies. This paper formalizes a class of problems of learning from ontology and data, and explores the design space of learning classifiers from attribute value taxonomies (AVTs) and data. We introduce the notion of AVT-extended data sources and partially specified data. We propose a general framework for learning classifiers from such data sources. Two instantiations of this framework, AVT-based Decision Tree classifier and AVT-based Naïve Bayes classifier are presented. Experimental results show that the resulting algorithms are able to learn robust high accuracy classifiers with substantially more compact representations than those obtained by standard learners.

1 Introduction

Current advances in machine learning have offered powerful approaches to exploring complex, a-priori unknown relationships or discovering hypotheses that describe potentially interesting regularities from data. Data-driven knowledge discovery in practice, occurs within a *context*, or under certain *ontological commitments* on the part of the learner. The learner's ontology (i.e., assumptions concerning *things* that exist in the *world*) determines the choice of *terms* and *relationships* among terms (or more generally, *concepts*) that are used to describe the domain of interest and their intended correspondence with objects and properties of the world [22]. This is particularly true in scientific discovery where specific ontological and representational commitments often reflect prior knowledge and working assumptions of scientists [8][27].

Hierarchical taxonomies over attribute values or classes are among the most common type of ontologies in practice. Examples of such ontologies include: Gene Ontology [3] that is a hierarchical taxonomy for describing many aspects of macromolecular sequence, structure and function; Hierarchical taxonomy built for features of intrusion detection [25]; Hierarchical groupings of attribute values for Semantic Web [5]; Hierarchies defined over data attributes in e-commerce applications of data mining [16].

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 308–321, 2005.

Making ontological commitments (that are typically implicit in a data set) *explicit* enables users to explore data from different points of view, and at different levels of abstraction. Each point of view corresponds to a set of ontological (and representational) commitments regarding the domain of interest. In scientific discovery, there is no single perspective that can serve all purposes, and it is always helpful to analyze data in different contexts and from alternative representations. Hence, there is a need for ontology-aware learning algorithms to facilitate the exploration of data from multiple points of view.

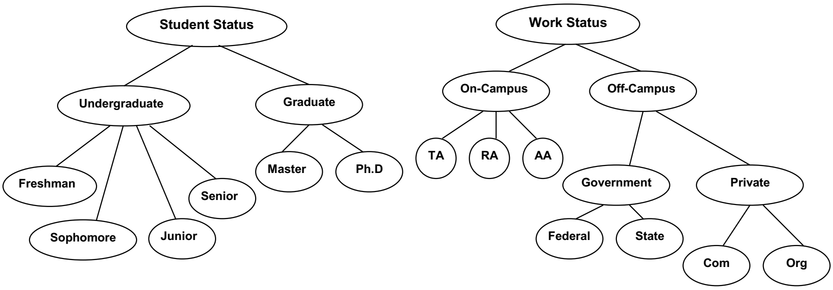
Exploring ontology-aware learning algorithms can provide us with a better understanding of the interaction between data and knowledge. The availability of user-supplied ontologies (e.g., taxonomies) presents the opportunity to learn classification rules that are expressed in terms of familiar hierarchically related concepts leading to simpler, easier-to-comprehend rules [26]. Moreover, learning algorithms that exploit hierarchical taxonomies can potentially perform a built-in regularization and thereby yielding robust classifiers [27].

Against this background, it is of significant practical interest to precisely formulate the problem of learning from ontologies (as a form of background knowledge or working assumptions) and data, and to explore the design space of algorithms for data-driven knowledge acquisition using *explicitly specified* ontologies (such as taxonomies). In this paper, we formalize the problem of learning pattern classifiers from Attribute Value Taxonomies, and propose a general learning framework that takes into account the tradeoff between the complexity and the accuracy of the predictive models. According to this general framework, we presents two well-founded AVT-based variants of machine learning algorithms, including Decision Tree and Naïve Bayes classifiers. We present our experimental results, and conclude with summary and discussion.

2 Problem Formulation

2.1 Ontology-Extended Data Source

In supervised classification learning problems, the data to be explored are typically available as a set of labelled training instances $\{(X_p, c_{X_p})\}$ where X_p is an instance in instance space I, and c_{X_p} is the class label from $C = \{c_1, c_2, \cdots, c_M\}$, a finite set of mutually disjoint classes. Assume that D is the data set represented using an ordered set of attributes $A = \{A_1, A_2, \cdots, A_N\}$, and $O = \{\mathit{\Lambda}_1, \mathit{\Lambda}_2, \cdots, \mathit{\Lambda}_N\}$ be an ontology associated with the data set. The element $\mathit{\Lambda}_i \in O$ corresponds to the attribute A_i, and describes the type of that particular attribute. In general, the type of an attribute can be a standard type (e.g., Integer or String) or a hierarchical type, which is defined as an ordering of a set of terms (e.g., attribute values). The schema S of the data set D is given by the set of attributes $\{A_1, A_2, \cdots, A_N\}$ used to describe the data together with their respective types $\{\mathit{\Lambda}_1, \mathit{\Lambda}_2, \cdots, \mathit{\Lambda}_N\}$ described by the ontology O. Caragea et al [8] defined *ontology-extended data source* to be expressed as $\mathcal{D} = \langle D, S, O \rangle$, where D is the data set, S is the schema of the data and O is the ontology



Student ID	Student Status	Work Status	Hourly Income	Internship
60-421	Freshman	Org	$10/hr.	No
73-727	Master	Com	$30/hr.	Yes
81-253	Ph.D	RA	$20/hr.	No
75-455	Graduate	On-Campus	$20/hr.	No
32-719	Sophomore	AA	$15/hr.	No
42-139	Senior	Government	$25/hr.	Yes
66-338	Undergraduate	Federal	$25/hr.	Yes
……	……	……	……	……

Fig. 1. Two attribute value taxonomies on student status and work status and a sample data set based on the two corresponding AVTs

associated with the data source. The instance space I where D is sampled can be defined as $I = \Lambda_1 \times \Lambda_2 \times \cdots \times \Lambda_N$

In the discussion that follows, we focus on hierarchical ontologies in the form of attribute value taxonomies (AVTs). Typically, attribute values are grouped into a hierarchical structure to reflect actual or assumed similarities among the attribute values in the domain of interest. We use $T = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N\}$ to represent the ordered set of attribute value taxonomies associated with attributes $A_1, A_2, \ldots, A_N$. Thus, an AVT defines an abstraction hierarchy over values of an attribute. Figure 1 shows an example of two AVTs, together with a sample data set collected by a university department based on the corresponding AVTs.

Specifically, we use *AVT-extended data source* $\mathcal{D} = \langle D, S, T\rangle$ to refer to the special case of ontology-extended data source where ontology is a set of attribute value taxonomies.

2.2 AVT-Induced Instance Space

In many real world application domains, the instances from AVT-extended data sources are often specified at different levels of precision. The value of a particular attribute or the class label associated with an instance or both are specified at different levels of abstraction with regard to the hierarchical taxonomies, leading to *partially specified instances* [27]. Partially specified data require us to extend our definition of instance space. We give formal definitions on partially specified data and AVT-induced instance space in the following.

Attribute value taxonomies enable us to specify a level of abstraction that reflects learner's perspective on the domain.

Definition 1 (Cut [14]). *A cut γ_i is a subset of elements in $Nodes(\mathcal{T}_i)$ satisfying the following two properties:* (1) *For any leaf $m \in Leaves(\mathcal{T}_i)$, either $m \in \gamma_i$ or m is a descendant of an element $n \in \gamma_i$; and* (2) *For any two nodes $f, g \in \gamma_i$, f is neither a descendant nor an ancestor of g.*

Definition 2 (Global Cut). *Let Δ_i be the set of all valid cuts in $\mathcal{T}_i$ of attribute A_i, and $\Delta = \times_{i=1}^{N} \Delta_i$ be the cartesian product of the cuts through the individual AVTs. $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_N\}$ defines a global cut through $T = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N\}$, where each $\gamma_i \in \Delta_i$ and $\Gamma \in \Delta$.*

Any global cut Γ in Δ specifies a level of abstraction for $\mathcal{D} = \langle D, S, T \rangle$. We use AVT frontier to refer to a global cut that is specified by the learning algorithm. In terms of a certain level of abstraction (i.e., a global cut Γ), we can precisely define fully specified instance and partially specified instance:

Definition 3 (Partially Specified Instance [27]). *If Γ represents the current level of abstraction in learner's AVT and $X_p = (v_{1p}, v_{2p}, ..., v_{Np})$ is an instance from D, then X_p is:*

- *Fully specified with respect to Γ, if $\forall i$, v_{ip} is on or below the cut Γ.*
- *Partially specified with respect to Γ, if $\exists v_{ip} \in X_p$, v_{ip} is above the cut Γ.*

When attribute value v_{ip} is below the specified cut Γ, it is fully specified because there is always a corresponding value on the cut that can replace the current value in the current level of abstraction. However, when v_{ip} is above the cut, there are several descendant values on the cut. It is uncertain which value will be the true attribute value, and hence partially specified. A particular attribute value can dynamically switch between being a fully specified value and being a partially specified value when the level of abstraction changes. For example, the shaded instances in Figure 1 are partially specified if the global cut Γ chooses to be all primitive attribute values in the corresponding AVTs.

The original instance space I is an instance space relative to a global cut Γ_ϕ with a domain of all primitive attribute values (all leaf-nodes in AVTs). Because any choice Γ defines a corresponding instance space I_Γ that is an abstraction of the original instance space I_{Γ_ϕ}, we can formally define AVT-induced instance space as follows.

Definition 4 (AVT-Induced Instance Space [28]). *A set of AVTs $T = \{T_1 \cdots T_N\}$ associated with a set of attributes $A = \{A_1 \cdots A_N\}$ induces an instance space $I_T = \cup_{\Gamma \in \Delta} I_\Gamma$ (the union of instance spaces induced by all of the cuts through the set of AVTs T).*

Therefore, a partially specified data set D_T is a collection of instances drawn from I_T where each instance is labeled with the appropriate class label from C. Thus, $D_T \subseteq I_T \times C$. Taking into account partially specified data, AVT-extended data source becomes $\mathcal{D} = \langle D_T, S, T \rangle$.

2.3 Learning Classifiers from Ontology-Extended Data Source

The problem of learning classifiers from data can be described as follows: Given a data set D, a hypothesis class H, and a performance criterion P, the classifier learner L generates a hypothesis in the form of a function $h : I \rightarrow C$, where $h \in H$ optimizes P. For example, we search for a hypothesis h that is most likely given the training data D.

Learning classifiers from an ontology-extended data set is a generalization of learning classifiers from data. The typical hypothesis class H has been extended to H_O, where the original hypothesis language has been enriched by ontology O. The resulting hypothesis space H_O is a much larger space. In the case where the ontology is a set of attribute value taxonomies, the hypothesis space changes to H_T, a collection of hypothesis classes $\{H_\Gamma | \Gamma \in \Delta\}$. Each H_Γ corresponds a hypothesis class with regard to a global cut Γ in the AVTs. Because partial ordering exists among global cuts, it is obvious that the resulting hypothesis space H_T also has partial ordering structure.

The problem of learning classifiers from AVT-extended data can be stated as follows: Given a user-supplied set of AVTs T and a data set D_T of (possibly) partially specified labeled instances, construct a classifier $h : I_T \rightarrow C$ for assigning appropriate class labels to each instance in the instance space I_T. It is the structure of the hypothesis space H_T that makes it possible to search the space efficiently for a hypothesis h that could be both concise and accurate.

3 AVT-Based Classifier Learners

We describe in the following a general framework for designing algorithms to learn classifiers from AVT-extended data sources. Base on this framework, we demonstrate our approach by extending standard decision tree classifier and Naïve Bayes classifier.

3.1 A General Learning Framework

There are essentially three elements in learning classifiers from AVT-extended data sources: (1) A procedure for identifying estimated sufficient statistics on AVTs from data; (2) A procedure for building and refining hypothesis; (3) A performance criterion for making tradeoff between complexity and accuracy of the generated classifiers. In what follows, we discuss each element in details.

(1) Identifying Estimated Sufficient Statistics

Building a classifier only needs certain *statistics* (i.e., a function of data). A statistic $\mathcal{S}(D)$ is called a *sufficient statistic* for a parameter θ if $\mathcal{S}(D)$ provides all the information needed for estimating the parameter θ from data D. We can formally define sufficient statistic for a learning algorithm.

Definition 5 (Sufficient Statistic for a Learning Algorithm [7]). *We say that $\mathcal{S}_L(D)$ is a sufficient statistic for learning the hypothesis h using a learning algorithm L if there exists a procedure that takes $\mathcal{S}_L(D)$ as input and outputs h.*

For many learning algorithms, sufficient statistics are frequency counts or class conditional frequency counts for attribute values. Given a hierarchical structured AVT, we can define a tree of frequency counts or class conditional frequency counts as the sufficient statistics for the learning algorithms. More specifically, with regard to an attribute value taxonomy $\mathcal{T}_i$ for attribute A_i, we define a tree of class conditional frequency counts $CCFC(\mathcal{T}_i)$ (and similarly, a tree of frequency counts $FC(\mathcal{T}_i)$).

If all the instances are fully specified in AVT-extended data source, the class conditional frequency counts associated with a non leaf node of $CCFC(\mathcal{T}_i)$ should correspond to the aggregation of the corresponding class conditional frequency counts associated with its children. $CCFC(\mathcal{T}_i)$ can be computed in one upward pass. When data are partially specified in AVT-extended data source, we can use a 2-step process for computing $CCFC(\mathcal{T}_i)$ [28]: First we make an upward pass aggregating the class conditional frequency counts based on the specified attribute values in the data set; Then we propagate the counts associated with partially specified attribute values down through the tree, augmenting the counts at lower levels according to the distribution of values along the branches based on the subset of the data for which the corresponding values are fully specified. This procedure can be seen as a special case of EM (Expectation Maximization) algorithm [11] to estimate sufficient statistics for $CCFC(\mathcal{T}_i)$.

(2) Building and Refining Hypothesis

As we have mentioned earlier, for a particular global cut Γ there is a corresponding hypothesis class H_Γ, and we can learn a hypothesis $h(\theta|\Gamma)$ with parameters θ from this hypothesis class H_Γ using a learning algorithm L. The total number of global cuts $|\Delta|$ grows exponentially with the scale of AVTs, and so does the number of possible hypotheses. Since an exhaustive search over the complete hypothesis space $\{H_\Gamma|\Gamma \in \Delta\}$ is computationally infeasible, we need a strategy to search through the resulting hypothesis space.

Following the definition of cut, we can define a refinement operation on cut as follows:

Definition 6 (Cut Refinement [28]). *We say that a cut $\hat{\gamma}_i$ is a refinement of a cut γ_i if $\hat{\gamma}_i$ is obtained by replacing at least one attribute value $v \in \gamma_i$ by its*

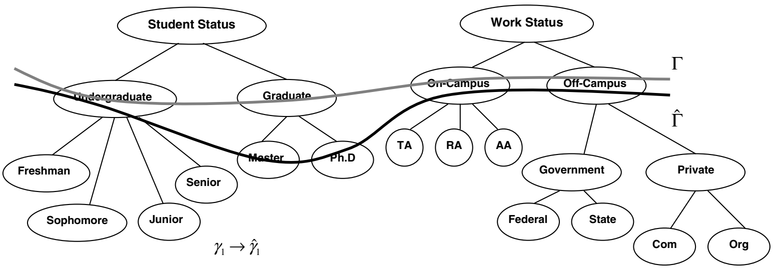


Fig. 2. Cut refinement. The cut γ_1={*Undergraduate*, *Graduate*} in the *student status* attribute has been refined to $\hat{\gamma_1}$={*Undergraduate*, *Master*, *Ph.D*}, and the global cut Γ has been refined to $\hat{\Gamma}$.

descendant attribute values. A global cut $\hat{\Gamma}$ is a refinement of a global cut Γ if at least one cut in $\hat{\Gamma}$ is a refinement of a cut in Γ.

Figure 2 shows a demonstrative cut refinement process based on the AVTs shown in Figure 1. When $\hat{\Gamma}$ is a cut refinement of Γ, the corresponding hypothesis $h(\hat{\Gamma})$ is a *hypothesis refinement* of $h(\Gamma)$. Hypothesis refinements in AVT-based learning are conducted through cut refinements in AVTs.

Based on gathered sufficient statistics, our goal is to search for the optimal hypothesis $h(\Gamma^*)$ from $\{H_\Gamma | \Gamma \in \Delta\}$, where Γ^* is an optimal level of abstraction (i.e., an optimal cut) that is decided by the learning algorithm L using certain performance measurement P.

We use a top-down refinement on the global cut to greedily explore the design space of the corresponding classifier. Our general strategy is to start by building a classifier that is based on the most abstract global cut and successively refine the classifier (hypothesis) by cut refinement. Therefore, the learning algorithm L generates a sequence of cut refinements $\Gamma_0, \Gamma_1, \cdots, \Gamma^*$, which corresponds to a sequence of hypothesis refinements $h(\Gamma_0), h(\Gamma_1), \cdots, h(\Gamma^*)$, until a final optimal cut Γ^* and an optimal classifier $h(\Gamma^*)$ is obtained.

(3) Trading Off the Complexity Against the Error

For almost every learning algorithm L, there is a performance measurement P that is explicitly or implicitly optimized by L. For example, some performance measurements include predictive accuracy, statistical significance tests, and many information criteria. However, the lack of good performance measurement makes the learning algorithm to build over complex model as the classifier that shows excellent performance on training data but poor performance on test data. This problem is called overfitting, which is a general problem that many learning algorithms seek to overcome.

Of particular interest to us are those criteria that can make tradeoffs between the accuracy and the complexity of the model [2][21], thereby having a built-in mechanism to overcome overfitting. For example, Minimum Description Length

(MDL) principle [21] is to compress the training data D and encode it by a hypothesis h such that it minimizes the length of the message that encodes both h and the data D given h. By making this tradeoff, we are able to learn classifiers that is both compact and accurate.

In order to perform hypothesis refinements effectively, we need a performance criterion P that can decide if we need to make a refinement from $h(\varGamma)$ to $h(\hat{\varGamma})$. Also this criterion should be able to decide whether we should stop making refinement and output a final hypothesis as the classifier.

The performance criterion P is applied in the calculation of sufficient statistics for hypothesis refinement that is defined as follows.

Definition 7 (Sufficient Statistics for Hypothesis Refinement[7]). *We denote $\mathcal{S}_L(D, h_i \rightarrow h_{i+1})$ as the sufficient statistic for hypothesis refinement from h_i to h_{i+1}, if the learner L accepts h_i and a sufficient statistic $\mathcal{S}_L(D, h_i \rightarrow h_{i+1})$ as inputs and outputs an updated hypothesis h_{i+1}.*

Different learning algorithms may use different performance criteria, and thus may have different formats and expressions of refinement sufficient statistics.

By combining the three elements of AVT-based classifier learners, we can write the following procedure to show this general learning framework.

1. Identify estimated sufficient statistics $\mathcal{S}_L(D)$ for AVTs as counts $\{CCFC(\mathcal{T}_i)\ |i = 1, ..., N\}$ or $\{FC(\mathcal{T}_i)|i = 1, ..., N\}$.
2. Initialize the global cut $\varGamma$ to the most abstract cut $\varGamma_0$.
3. Based on the estimated sufficient statistic, generate a hypothesis $h(\varGamma)$ corresponding to the current global cut $\varGamma$ and learn its parameters.
4. Generate a cut refinement $\hat{\varGamma}$ on $\varGamma$, and construct hypothesis $h(\hat{\varGamma})$
5. Calculate $\mathcal{S}_L(D, h(\varGamma) \rightarrow h(\hat{\varGamma}))$ for hypothesis refinement from $h(\varGamma)$ to $h(\hat{\varGamma})$.
6. Based on performance criterion P, if stopping criterion is met, then output $h(\varGamma)$ as the final classifier; else if the condition for hypothesis refinement is met, set current hypothesis to $h(\hat{\varGamma})$ by replacing $\varGamma$ with $\hat{\varGamma}$, else keep $h(\varGamma)$, and goto step 4;

Next, we discuss two instantiations of this learning framework and identify their corresponding elements within the same framework.

3.2 AVT-Based Naïve Bayes Learner (AVT-NBL)

AVT-NBL [28] is an extension of the standard Naïve Bayes learning algorithm that effectively exploits user-supplied AVTs to construct compact and accurate Naïve Bayes classifier from partially specified data. We can easily identify the three elements in the learning framework for AVT-NBL as follows:

(1) The sufficient statistics $\mathcal{S}_L(D)$ for AVT-NBL is the class conditional frequency counts $\{CCFC(\mathcal{T}_i)|i = 1, ..., N\}$.

(2) The hypothesis refinements strictly follow the procedure of cut refinements in the framework. When a global cut Γ is specified, there is a corresponding Naïve Bayes classifier $h(\Gamma)$ that is completely specified by a set of class conditional probabilities for the attribute values on Γ. Because each attribute is assumed to be independent of others given the class, the search for the AVT-based Naïve Bayes classifier (AVT-NBC) can be performed efficiently by optimizing the criterion independently for each attribute.

(3) The performance criterion that AVT-NBL optimizes is the Conditional Minimum Description Length (CMDL) score suggested by Friedman et al [12]. CMDL score can be calculated as follows:

$$CMDL(h(\Gamma)|D) = \left(\frac{\log|D|}{2}\right) size(h(\Gamma)) - CLL(h(\Gamma)|D)$$

$$where,\ CLL(h(\Gamma)|D) = |D|\sum_{p=1}^{|D|} \log P_h(c_{X_p}|v_{1p}, \cdots, v_{Np})$$

where, $P_h(c_{X_p}|v_{1p}, \cdots, v_{Np})$ is the class conditional probability, $size(h(\Gamma))$ is the number of parameters used by $h(\Gamma)$, $|D|$ the size of the data set, and $CLL(h(\Gamma)|D)$ is the conditional log likelihood of the hypothesis $h(\Gamma)$ given the data D. In the case of a Naïve Bayes classifier, $size(h(\Gamma))$ corresponds to the total number of class conditional probabilities needed to describe $h(\Gamma)$.

The sufficient statistics for hypothesis refinement in AVT-NBL can be quantified by the difference between their respective CMDL scores: $s_L(D, h(\Gamma) \rightarrow h(\hat{\Gamma})) = CMDL(h(\hat{\Gamma})|D) - CMDL(h(\Gamma)|D)$. If $s_L(D, h(\Gamma) \rightarrow h(\hat{\Gamma})) > 0$, $h(\Gamma)$ is refined to $h(\hat{\Gamma})$. This refinement procedure terminates when no further refinement can make improvement in the CMDL score (i.e., the stoping criterion).

3.3 AVT-Based Decision Tree Learner (AVT-DTL)

AVT-DTL [27] implements a top-down AVT-guided search in decision tree hypothesis space, and is able to learn compact and accurate decision tree classifier from partially specified data. Similarly, we can identify the three elements in the learning framework for AVT-DTL as follows:

(1) The sufficient statistics $\mathcal{S}_L(D)$ for AVT-DTL is the frequency counts $\{FC(\mathcal{T}_i)|i = 1, ..., N\}$.

(2) The hypothesis refinement is incorporated into the process of decision tree construction. The cut refinement is done by keeping track of "pointing vectors" in the AVTs. Each "pointing vector" is a set of pointers, and each pointer points to a values in an AVT. As an example, in Figure 3, the pointing vector points to two high-level attribute values in the two corresponding taxonomies.

The union of the set of pointing vectors at all leaves of a partially constructed decision tree corresponds to a global cut in AVTs. Obviously, any global cut in the constructed decision tree has a corresponding global cut in AVTs. At each stage of decision tree construction, we have a current set of pointing vectors as the global cut Γ being explored, and a corresponding partially constructed decision tree to be the hypothesis $h(\Gamma)$. AVT-DTL indirectly makes refinement on Γ by updating each pointing vector, and hence makes hypothesis refinement on $h(\Gamma)$ and grows the decision tree accordingly. AVT-DTL does not have the

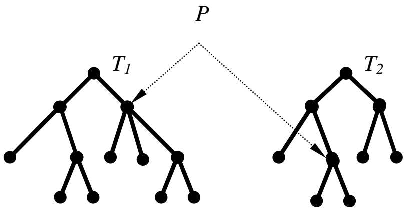


Fig. 3. Illustration of a Pointing Vector P

independent assumption on attributes given the class, the search is conducted globally to make refinements on possible cuts.

(3) The performance criterion that AVT-DTL uses is the standard information gain or gain ratio [20]. The sufficient statistic for hypothesis refinement is exactly the information criterion: $s_L(D, h(\Gamma) \rightarrow h(\hat{\Gamma})) = info(\Gamma \rightarrow \hat{\Gamma})$, where $info(\Gamma \rightarrow \hat{\Gamma})$ is the information gain (or gain ratio) when current decision tree $h(\Gamma)$ has been extended to $h(\hat{\Gamma})$. The stopping criterion for AVT-DTL is the same for standard decision tree. For example, such stopping criterion can be χ^2 test to test statistical significance on further split.

4 Experiments and Results

We summarize below, results of experiments that compare the performance of standard learning algorithm (DTL, NBL) with that of their AVT-based counterparts (AVT-DTL/AVT-NBL) as well as the standard learning algorithms applied to a propositionalized version of the data set (PROP-DTL/PROP-NBL) [27]. In propositionalized method, the data set is represented using a set of Boolean

Table 1. Comparison of error rate and size of classifier generated by NBL, PROP-NBL and AVT-NBL on benchmark data

% Error rates using 10-fold cross validation with 90% confidence interval; The size of the classifiers for each data set is constant for NBL and Prop-NBL, and for AVT-NBL, the size shown represents the average across the 10-cross validation experiments.

DATA SET	NBL		PROP-NBL		AVT-NBL	
	ERROR	*SIZE*	ERROR	*SIZE*	ERROR	*SIZE*
Audiology	26.55 (±5.31)	*3696*	27.87 (±5.39)	*8184*	23.01 (±5.06)	*3600*
Breast-Cancer	28.32 (±4.82)	*84*	27.27 (±4.76)	*338*	27.62 (±4.78)	*62*
Car	14.47 (±1.53)	*88*	15.45 (±1.57)	*244*	13.83 (±1.50)	*80*
Dermatology	2.18 (±1.38)	*876*	1.91 (±1.29)	*2790*	2.18 (±1.38)	*576*
Mushroom	4.43 (±1.30)	*252*	4.45 (±1.30)	*682*	0.14 (±0.14)	*202*
Nursery	9.67 (±1.48)	*135*	10.59 (±1.54)	*355*	9.67 (±1.48)	*125*
Soybean	7.03 (±1.60)	*1900*	8.19 (±1.72)	*4959*	5.71 (±1.45)	*1729*
Zoo	6.93 (±4.57)	*259*	5.94 (±4.25)	*567*	3.96 (±3.51)	*245*

Table 2. Comparison of error rate and size of classifier generated by C4.5, PROP-C4.5 and AVT-DTL on benchmark data No printing. No pruning is applied.

% Error rates using 10-fold cross validation with 90% confidence interval; The size of the classifier for each data set represents the average across the 10-cross validation experiments.

DATA SET	C4.5		PROP- C4.5		AVT- DTL	
	ERROR	*SIZE*	ERROR	*SIZE*	ERROR	*SIZE*
Audiology	23.01 (±5.06)	*37*	23.01 (±5.06)	*26*	21.23 (±4.91)	*30*
Breast-Cancer	33.91 (±5.06)	*152*	32.86 (±5.03)	*58*	29.37 (±4.87)	*38*
Car	7.75 (±1.16)	*297*	1.79 (±0.58)	*78*	1.67 (±0.57)	*78*
Dermatology	6.83 (±2.38)	*71*	5.74 (±2.20)	*19*	5.73 (±2.19)	*22*
Mushroom	0.0 (±0.00)	*26*	0.0 (±0.00)	*10*	0.0 (±0.00)	*10*
Nursery	3.34 (±0.90)	*680*	1.75 (±0.66)	*196*	1.21 (±0.55)	*172*
Soybean	9.81 (±2.06)	*175*	8.20 (±1.90)	*67*	7.75 (±1.85)	*90*
Zoo	7.92 (±4.86)	*13*	8.91 (±5.13)	*9*	7.92 (±4.86)	*7*

Table 3. Comparison of error rate on data with 10%, 30% and 50% partially or totally missing values. The error rates were estimated using 10-fold cross validation, and we calculate 90% confidence interval on each error rate.

DATA		PARTIALLY MISSING			TOTALLY MISSING		
METHODS		NBL	PROP-NBL	AVT-NBL	NBL	PROP-NBL	AVT-NBL
MUSHROOM	10%	4.65(±1.33)	4.69(±1.34)	0.30(±0.30)	4.65(±1.33)	4.76(±1.35)	1.29(±071)
	30%	5.28 (±1.41)	4.84(±1.36)	0.64(±0.50)	5.28 (±1.41)	5.37(±1.43)	2.78(±1.04)
	50%	6.63(±1.57)	5.82(±1.48)	1.24(±0.70)	6.63(±1.57)	6.98(±1.61)	4.61(±1.33)
NURSERY	10%	15.27(±1.81)	15.50(±1.82)	12.85(±1.67)	15.27(±1.81)	16.53(±1.86)	13.24(±1.70)
	30%	26.84(±2.23)	26.25(±2.21)	21.19(±2.05)	26.84(±2.23)	27.65(±2.24)	22.48(±2.09)
	50%	36.96(±2.43)	35.88(±2.41)	29.34(±2.29)	36.96(±2.43)	38.66(±2.45)	32.51(±2.35)
SOYBEAN	10%	8.76(±1.76)	9.08(±1.79)	6.75(±1.57)	8.76(±1.76)	9.09(±1.79)	6.88(±1.58)
	30%	12.45(±2.07)	11.54(±2.00)	10.32(±1.90)	12.45(±2.07)	12.31(±2.05)	10.41(±1.91)
	50%	19.39(±2.47)	16.91(±2.34)	16.93(±2.34)	19.39 (±2.47)	19.59(±2.48)	17.97(±2.40)

attributes obtained from $\mathcal{T}_i$ of attribute A_i by associating a Boolean attribute with each node (except the root) in $\mathcal{T}_i$. Thus, each instance in the original data set defined using N attributes is turned into a Boolean instance specified using $\tilde{N}$ Boolean attributes where $\tilde{N} = \sum_{i=1}^{N}(|Nodes(\mathcal{T}_i)| - 1)$.

The data sets used in our experiments [27][28] were based on benchmark data sets available in the UC-Irvine repository. AVTs were supplied by domain experts for some of the data sets. For the remaining data sets, the AVTs were generated using AVT-Learner, a Hierarchical Agglomerative Clustering (HAC) algorithm for constructing AVTs [15].

Table 1 shows the estimated error rates of the Naïve Bayes classifiers generated by the AVT-NBL, NBL, and PROP-NBL on benchmark data sets [28].

Table 2 shows the estimated error rates of the decision tree classifiers generated by the AVT-DTL, C4.5 [20], and PROP-C4.5 on the same benchmark data sets.

Experiments were also run with synthetic data sets with different prespecified percentages of totally or partially missing attribute values generated from the original benchmark data sets. Table 3 compares the estimated error rates of AVT-NBL with that of NBL and PROP-NBL in the presence of varying percentages of partially missing attribute values and totally missing attribute values [28].

Our main results can be summarized as follows: (1) AVT-DTL and AVT-NBL are able to learn robust high accuracy classifiers from data sets consisting of partially specified data comparing to those produced by their standard counterparts on original data and propositionalized data. (2) Both AVT-DTL and AVT-NBL yield substantially more compact and comprehensible classifiers than standard version and propositionalized version of standard classifiers.

5 Summary and Discussion

5.1 Summary

Ontology-aware classifier learning algorithms are needed to explore data from multiple points of view, and to understand the interaction between data and knowledge. By exploiting ontologies in the form of attribute value taxonomies in learning classifiers from data, we are able to construct robust, accurate and easy-to-comprehend classifiers within a particular domain of interest.

We provide a general framework for learning classifiers from attribute value taxonomies and data. We illustrate the application of this framework in the case of AVT-based variants of decision tree and Naïve Bayes classifiers. However, this framework can be used to derive AVT-based variants of other learning algorithms, such as nonlinear regression classifiers, support vector machines, etc.

5.2 Related Work

Several authors have explored the use of attribute value taxonomies in learning classifiers from data. [1][9][10][13][17][23][27][28]. The use of prior knowledge or domain theories specified in first order logic or propositional logic to guide learning from data has been explored in ML-SMART [4], FOCL [19], and KBANN [24] systems. However, the work on exploiting domain theories in learning has not focused on the effective use of AVT to learn classifiers from partially specified data. McClean et al [18] proposed aggregation operators defined over partial values in databases. Caragea et al have explained the use of ontologies in learning classifiers from semantically heterogeneous data [8]. The use of multiple independent sets of features has led to "multi-view" learning [6]. However, our work focuses on exploring data with associated AVTs at multiple levels of abstraction, which corresponds to multiple points of view of the user.

In this paper, we have described a general framework for deriving ontology-aware algorithms for learning classifiers from data when ontologies take the form of attribute value taxonomies.

5.3 Future Work

Some promising directions for future work in ontology-guided data-driven learning include:

(1) Design of AVT-based variants of other machine learning algorithms. Specifically, it would be interesting to design AVT and CT-based variants of algorithms for construction Bag-of-words classifiers, Bayesian Networks, Nonlinear Regression Classifiers, and Hyperplane classifiers (Perceptron, Winnow Perceptron, and Support Vector Machines).

(2) Extensions that incorporate richer classes of AVT. Our work has so far focused on tree-structured taxonomies defined over nominal attribute values. It would be interesting to extend this work in several directions motivated by the natural characteristics of data: (a) Hierarchies of Intervals to handle numerical attribute values; (b) Ordered generalization Hierarchies where there is an ordering relation among nodes at a given level of a hierarchy (e.g., hierarchies over education levels); (c) Tangled Hierarchies that are represented by directed acyclic graphs (DAG) and Incomplete Hierarchies which can be represented by a forest of trees or DAGs.

Acknowledgments

This research was supported in part by grants from the National Science Foundation (NSF IIS 0219699) and the National Institutes of Health (GM 066387).

References

1. Almuallim H., Akiba, Y., Kaneda, S.: On Handling Tree-Structured Attributes. Proceedings of the Twelfth International Conference on Machine Learning (1995)
2. Akaike, H.: A New Look at Statistical Model Identification. IEEE Trans. on Automatic Control, AU-19:716-722. (1974)
3. Ashburner, M., et al.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet. 25(1). (2000)
4. Bergadano, F., Giordana, A.: Guiding Induction with Domain Theories. In: Machine Learning - An Artificial Intelligence Approach. Vol. 3, pp 474-492, Morgan Kaufmann. (1990)
5. Berners-Lee, T., Hendler, J. and Lassila, O.: The semantic web. Scientific American, May. (2001)
6. Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-Training. Annual Conference on Computational Learning Theory. (COLT-1998)
7. Caragea, D., Silvescu, A., and Honavar, V.: A Framework for Learning from Distributed Data Using Sufficient Statistics and its Application to Learning Decision Trees. International Journal of Hybrid Intelligent Systems. Vol. 1 (2004)
8. Caragea, D., Pathak, J., and Honavar, V.: Learning Classifiers from Semantically Heterogeneous Data. In 3rd International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems. (2004)

9. A. Clare, R. King.: Knowledge Discovery in Multi-label Phenotype Data. In: Lecture Notes in Computer Science. Vol. 2168. (2001)
10. W. Cohen.: Learning Trees and Rules with Set-valued Features. In. Proceedings of the Thirteenth National Conference on Artificial Intelligence. AAAI Press, (1996)
11. Dempster A., Laird N., Rubin D.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, Series B, 39(1), pp 1-38. (1977)
12. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian Network Classifiers. Machine Learning, Vol: 29. (1997)
13. Han, J., Fu, Y.: Exploration of the Power of Attribute-Oriented Induction in Data Mining. U.M. Fayyad, el al. (eds.), Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press. (1996)
14. Haussler, D.: Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework. Artificial Intelligence, 36. (1988)
15. D. Kang, A. Silvescu, J. Zhang, and V. Honavar. Generation of Attribute Value Taxonomies from Data for Data-Driven Construction of Accurate and Compact Classifiers. To appear: Proceedings of The Fourth IEEE International Conference on Data Mining, 2004.
16. Kohavi, R., Provost, P.: Applications of Data Mining to Electronic Commerce. Data Mining and Knowledge Discovery, Vol. 5. (2001)
17. D. Koller, M. Sahami.: Hierarchically classifying documents using very few words. In: Proceedings of the 14th Int'l Conference on Machine Learning. (1997)
18. McClean S., Scotney B., Shapcott M.:Aggregation of Imprecise and Uncertain Information in Databases. IEEE Trans. on Knowledge and Data Engineering Vol. 13(6), pp 902-912. (2001)
19. Pazzani M., Kibler D.: The role of prior knowledge in inductive learning. Machine Learning, Vol. 9, pp 54-97. (1992)
20. Quinlan, J. R.: C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufmann (1992)
21. Rissanen, J.: Modeling by shortest data description. Automatica, vol. 14. (1978)
22. Sowa, J.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. New York: PWS Publishing. (1999)
23. Taylor, M., Stoffel, K., Hendler, J.: Ontology-based Induction of High Level Classification Rules. SIGMOD Data Mining and Knowledge Discovery workshop proceedings. Tuscon, Arizona (1997)
24. Towell, G., Shavlik, J.: Knowledge-based Artificial Neural Networks. Artificial Intelligence, Vol. 70. (1994)
25. Undercoffer, J., et al.: A Target Centric Ontology for Intrusion Detection: Using DAML+OIL to Classify Intrusive Behaviors. To appear, Knowledge Engineering Review - Special Issue on Ontologies for Distributed Systems, Cambridge University Press. (2004)
26. Zhang, J., Silvescu, A., Honavar, V.: Ontology-Driven Induction of Decision Trees at Multiple Levels of Abstraction. Proceedings of Symposium on Abstraction, Reformulation, and Approximation. Lecture Notes in Computer Science 2371. (2002)
27. Zhang, J., Honavar, V.: Learning Decision Tree Classifiers from Attribute Value Taxonomies and Partially Specified Instances. In: Proceedings of the 20th Int'l Conference on Machine Learning. (2003)
28. Zhang, J., Honavar, V.: AVT-NBL: An Algorithm for Learning Compact and Accurate Naïve Bayes Classifiers from Attribute Value Taxonomies and Data. In: Proceedings of the Fourth IEEE International Conference on Data Mining. (2004)

Automatic Extraction of Proteins and Their Interactions from Biological Text*

Kiho Hong[1], Junhyung Park[2], Jihoon Yang[2], and Eunok Paek[3]

[1] IT Agent Research Lab, LSIS R&D Center,
Hogae-dong, Dongsan-Gu, Anyang-Shi, Kyungki-Do 431-080, Korea
khhong1@lsis.biz

[2] Department of Computer Science and
Interdisciplinary Program of Integrated Biotechnology, Sogang University,
1 Shinsoo-Dong, Mapo-Ku, Seoul 121-742, Korea
jhpark@mllab.sogang.ac.kr, yangjh@sogang.ac.kr

[3] Department of Mechanical and Information Engineering, The University of Seoul,
90 Jeonnong-Dong, Dongdaemun-Gu, Seoul 130-743, Korea
paek@uos.ac.kr

Abstract. Text mining techniques have been proposed for extracting protein names and their interactions from biological text. First, we have made improvements on existing methods for handling single word protein names consisting of characters, special symbols, and numbers. Second, compound word protein names are also extracted using conditional probabilities of the occurrences of neighboring words. Third, interactions are extracted based on Bayes theorem over discriminating verbs that represent the interactions of proteins. Experimental results demonstrate the feasibility of our approach with improved performance in terms of accuracy and F-measure, requiring significantly less amount of computational time.

1 Introduction

In biologically significant applications such as developing a new drug and curing an inveterate disease, understanding the mutual effects of proteins (or genes which will be used interchangeably in the paper) are essential [1]. For instance, in order to develop a medicine for the breast cancer, we need to figure out the proteins related to the disease, and understand the mechanism how they work together in the course of the development of the breast cancer. In order to achieve the goal, extracting gene names must be proceeded. However, results by some of the existing methods leave much to be desired (e.g. extraction of multiple protein names, handling of negative and compound sentences and special characters). [2,3]. Motivated by this background, we propose a new approach to extracting gene names and their relations. Section 2 and 3 describe the extraction of gene names and interactions between them. Section 4 shows experimental results in comparison with other approaches, followed by concluding remarks in Section 5.

* This work was supported by grant No. R01-2004-000-10689-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 322–329, 2005.

2 Extraction of Protein Names

A Protein is named either as a single word (i.e. singular protein name (SPN)) or multiple words (i.e. multiple protein name (MPN)). We describe extracting methods for each case.

2.1 SPN Extraction

A SPN is extracted by two steps:

1. Word Class Tagging
   First, we used the Brill's tagger for tagging the text [4]. We added a word class GENE and prepared a list of the words in the class. GenBank[1] database is used for making the list. To define lexicon rules and context rules during the tagger's learning stage, we used GENIA CORPUS [2,5].
2. SPN Extraction
   Generally, the protein names in biological literature are usually irregular and ambiguous. Even though there exist some rules for protein naming (some can be found at Nature Genetics site [3]), it is hard to apply the rules to existing protein names. Also as the rules are not generalized, some of the special characters are used frequently (e.g. hyphens, Greek letters, digits and Roman letters). In our lexicon, about 37% of these special characters are contained in the text. For this reason, processing them plays a great role for the whole efficiency. The HMM(Hidden Markov Model) with the Viterbi algorithm is applied for SPN extraction [6]. In addition to the algorithm, in order to handle the special characters, a substitution method was considered (e.g. & for digits and ? for roman letters). Substring matching was applied to the substituted protein names. However, there could be a collision in substring matching. For instance, 'gap1' will be substituted to 'gap&' which can be confused with 'gap' since 'gap' and 'gap&' has the same prototype 'gap'. Therefore, a set of words that can be confused in this fashion has been reserved as stopwords, which are ignored.

2.2 MPN Extraction

Usually an SPN makes up an MPN with near (or neighboring) words. However, an MPN not including an SPN should be considered as well (e.g. tumor necrosis factor). Based on the technique used in TagGeN [3], we developed an enhanced probability model. First, if GENE tag is included, the range of an MPN is determined by expanding words in bidirection (i.e. right and left). If an MPN does not include any GENE word, we use SEED word (e.g. the words appearing in MPNs frequently) for MPN determination. In our experiment, about 80 SEED words were used. To determine the range of an MPN, it is needed to expand the search from a GENE word or a SEED word, considering the following probability:

$$P(W_{next}|W_{current}, M_{current} = 1) \qquad (1)$$

[1] http://www.ncbi.nlm.nih.gov/Genbank/index.html

Table 1. Examples of W_i's Used in Probabilistic Models

Left Direction	
Set	*Example*
NN(Noun Class)	single-chain/**NN** fv/GENE
JJ(Adjective Class)	human/**JJ** GM-CSF/**GENE** gene/NN
CD(Number Class)	3/**CD** alpha/NN HSD/GENE
GENE(Gene Class)	human/JJ GM-CSF/**GENE** gene/NN
...ase	phospholip**ase**
Roman, Greek Character	type **II** IL-1R
Word Set(i.e. protein, gene, factor, etc.)	**protein** tyrosine kinase
Right Direction	
Set	*Example*
reporter	beta-globin **reporter**
product	start-1 gene **product**
single character	c-erb **A**
Numerals	IFN-stimulated gene factor **3**
...ed	C5a induc**ed** kappa-B
...like	Proximal c-jun TRE-**like** promoter element
...ing	IRF-1 GAS-bind**ing** complex

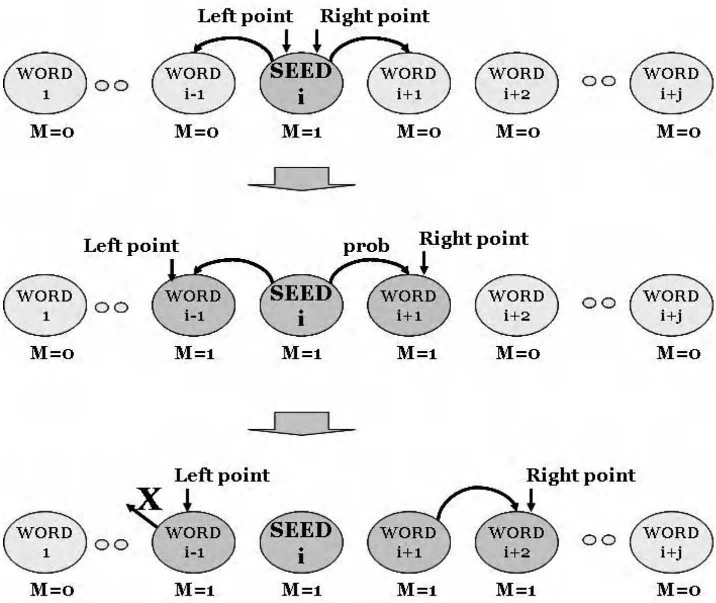


Fig. 1. Probabilistic Model for MPN Tagging

where W_i represents a word occurring at position i, and M_i is binary value which represents whether the word at position i belongs to GENE word class or not. Some of the examples of W_i are illustrated in Table 1.

Initially, only M values of SEED words have 1 and all the others have 0. From the SEED word in the middle of the MPN, we move bidirectionally (i.e. to the right and to the left). By calculating the probability in (1), we calculate M values which represent whether the word is included in the MPN or not. Generally, the left-hand side words of an MPN have diverse word classes than the right ones, and the right-hand side words of an MPN consist of Greek letters, Roman letters and digits. This bidirectional expansion of words is expected to generate a more accurate model than that by TagGeN [3]. In order to make a probabilistic model for MPN extraction, we used 600 documents which are arbitrarily chosen from GENIA corpus and pre-tagged by domain experts. Figure 1 illustrates the probabilistic model used for tagging MPN from documents.

3 Extraction of Protein Interactions

This section describes the method for protein interactions. For example, there could be a pattern like *'Protein(A)-Type(interaction)-Protein(B)'* [4]. We define the verbs for the interactions and extract events from these predefined patterns. Then we are able to know that entity A has a relation with B. We first extract the discriminating verbs and then extract the associated protein interactions.

3.1 Discriminating Verb Extraction

A discriminating verb is extracted as follows:

1. Pre-processing
   The set of types (i.e. interactions) we are interested in would be the discriminating verb set. To define the set, pre-processing for extracting verbs from the text is needed. This can be done easily as Brill's tagger tags verbs as VB(verb, base form) including VBN(verb, past participle), and VBZ(verb, 3rd person singular present) that we can extract and stem.
2. P-Score Estimation
   We design a Bayesian probabilistic model for estimating the P-Score of each verb in the document. Then, we determine the set of discriminating verbs based on the P-Scores. The P-Score exhibits how well a verb describes the interaction between proteins. This was proposed for extracting a word set to classify documents by Marcotte [7]. We applied the method for extraction of discriminating verbs and calculate the following probability:

   $$P(n|N,f) \approx e^{-Nf}\frac{(Nf)^n}{n!} \tag{2}$$

   where n means how many times a verb is used as a protein interaction, N is total number of words in a document, and f is the total occurrences of each verb. The Poisson distribution can be an alternative for $P(n|N,f)$ while N is big enough and f is fairy small.

3. Discriminating Verb Selection
   Calculate the P-Score for every word, and then choose a set of arbitrary number of words with the highest P-Scores. 80 words (e.g. inhibit, indicate, etc.) were used in our experiment.

3.2 Protein Mutual Effect Extraction

To extract an interaction between genes from a sentence, there should be more than two gene names and one verb which describes their relation. However, due to an ambiguity of natural language, it is hard to recognize the structure well. We introduce a simple method to decrease the ambiguity of natural language structures. The steps of extracting protein interaction by using discriminating verbs and events are as follows:

1. Complex Sentence Processing
   To handle the ambiguity in a sentence, we used Toshihide Ono's method [1]. The method diminishes the ambiguity by converting a complex sentence into simple sentences and a negative sentence into a positive one.
2. Interaction Extraction
   If there is a pattern like *'Protein(A)-Type(Verb)- Protein(B)'* and a discriminating verb in a sentence, we calculate $Confidence$ of the sentence and then add the sentence into the *event* (protein interaction) set.

The *Confidence* is calculated as follows:

$$Confidence = s + \frac{1}{sd} \tag{3}$$

where s is a binary value which represents whether the pattern is included in the sentence or not, and sd is sum of distances from proteins to a verb in the sentence. The distance is a number of words from a verb to proteins in a sentence. For example, 'IL-10 inhibits IFN-gamma-induced ICAM-1 expression in monocytes.' has distance 2 as *IL-10* and *inhibit* have distance 1 and *inhibit* and *IFN-gamma-induced ICAM-1 expression* have distance 1, too.

A sentence with no discriminating verb is also added to the candidate event set. We re-calculate $Confidence$ with $Frequency$ (how many times protein(A) and (B) are found in documents).

4 Experiments

We obtained the following extraction results of proteins and their interactions. Data used for the experiments are 600 papers from the GENIA Corpus. Our results are compared with those by ABGene and TagGeN [2,3] in following tables.

- SPN Extraction
  To observe the results while a data set size is changing, we experimented on 100 to 600 documents. Table 2 exhibits comparable accuracies among

Table 2. Accuracy of SPN Extraction

Dataset System	100	200	300	400	500	600	Average
Our system	83.28	85.17	84.97	85.10	85.58	85.88	85.00(%)
ABGene	87.40	87.12	87.13	87.19	86.12	87.10	87.01(%)
TagGeN	80.17	82.24	83.51	84.09	84.50	84.91	83.24(%)

Table 3. Recall of SPN Extraction

Dataset System	100	200	300	400	500	600	Average
Our system	95.06	95.99	96.39	97.00	96.89	96.33	96.27(%)
ABGene	50.15	57.75	49.16	54.12	60.02	61.12	55.22(%)
TagGeN	68.75	75.49	78.32	77.16	78.82	79.09	76.27(%)

Table 4. F-measure of SPN Extraction

Dataset System	100	200	300	400	500	600	Average
Our system	88.78	90.26	90.32	90.66	90.88	90.80	90.28(%)
ABGene	63.74	69.02	62.86	66.79	70.74	71.83	67.56(%)
TagGeN	74.02	78.72	80.83	80.48	81.56	81.90	79.56(%)

Table 5. Processing Time of SPN Extraction

Dataset System	100	200	300	400	500	600
Our system	2.81	3.50	4.23	4.85	5.46	6.23(sec)
ABGene	19.01	39.28	56.12	74.31	94.11	113.00(sec)
TagGeN	5913	11925	18777	24970	30979	36324(sec)

the approaches, with no conspicuous differences in performance for various sizes of data. Due to the substring matching method our system showed 2% low accuracy than that of ABGene, while it produced high recall and F-measure as shown in Table 3 and Table 4. In addition, our system was order of magnitude faster due to protein name hashing and simplified tagging process, as shown in Table 5.

– MPN Extraction
‘Exact’ means the case every words in an MPN is extracted correctly. When some range of an MPN is partially extracted, it is named as ‘Partial’. As shown in Table 6, our approach outperformed TagGeN in MPN extraction.

Table 6. Performance of MPN Extraction

	Recall(%)	Precision(%)	F-measure(%)
Our system(exact/partial)	84.25/91.56	86.65/91.35	84.84/91.84
TagGeN(exact/partial)	80.23/86.51	87.81/91.15	83.84/88.77

Table 7. Performance of Interaction Extraction

Precision(%)	Recall(%)	F-measure(%)
76.58	92.70	83.87

– Protein Interaction Extraction
  We used 80 discriminating verbs in order of high P-Score. Selected 100 sentences including 14 negative, 8 compound sentence structures, and 121 protein interactions were used. From the sentences, we got 139 protein interactions. The number of interactions obtained by only discriminating verbs were 89, and 50 relations were added from the sentences in the candidate event set. We obtained F-measure over 80% as shown in Table 7.

5 Conclusion

We developed an extraction system for proteins and their interactions. Our protein name substring matching method and more abundant lexicon improved overall system performance. We also defined discriminating verbs and extracted them using a probabilistic model. We extracted 80 discriminating verbs by Poisson distribution. Finally, we defined events, and by their confidence values extracted their interactions. We observed improved performance in experiments with biological data.

Some of future research directions include: First, current simple substring matching method might cause low precision, which can be improved; Second, current algorithm includes ad hoc steps, and a more systematic algorithm for interaction extraction can be devised; Third, a thoughtful consideration for natural language processing is needed for more enhanced information extraction; Finally, more experiments with additional data will help verify our system.

References

1. Ono, T., Hishigaki, H., Tanigami, A., Takagi, T.: Automated extraction of information on protein-protein interactions from the biological literature. Bioinformatics **17** (2001) 155–161
2. Kim, J.D., Ohta, T., Tateisi, Y., Tsujii, J.: Genia corpus - a semantically annotated for bio-textmining. Bioinformatics **19** (2002) 180–192
3. Tanabe, L., Wilbur, W.J.: Tagging gene and protein names in full text article. In: Proceedings of Association for Computational Linguistics. (2004) 9–13

4. Brill, E.: Some advances in transformation-based part of speech tagging. In: AAAI. (1994)
5. Rinaldi, F., Schneider, G., Kaljurand, K., Dowda'll, J., Andronis, C., Persidis, A., Konstanti, O.: Mining relations in the genia corpus. In: Proceedings of the Second European Workshop and Text mining for Bioinformatics. (2004)
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. second edn. Wiley-interscience. Inc. (2000)
7. Marcotte, E.M., Xenarios, I., Eisenberg, D.: Mining literature for protein-protein interactions. Bioinformatics **17** (2002) 359–363

A Data Analysis Approach for Evaluating the Behavior of Interestingness Measures

Xuan-Hiep Huynh, Fabrice Guillet, and Henri Briand

LINA CNRS 2729 - Polytechnic School of Nantes University,
La Chantrerie BP 50609 44306 Nantes cedex 3, France
{xuan-hiep.huynh, fabrice.guillet, henri.briand}@polytech.univ-nantes.fr

Abstract. In recent years, the problem of finding the different aspects existing in a dataset has attracted many authors in the domain of knowledge quality in KDD. The discovery of knowledge in the form of association rules has become an important research. One of the most difficult issues is that an enormous number of association rules are discovered, so it is not easy to choose the best association rules or knowledge for a given dataset. Some methods are proposed for choosing the best rules with an interestingness measure or matching properties of interestingness measure for a given set of interestingness measures. In this paper, we propose a new approach to discover the clusters of interestingness measures existing in a dataset. Our approach is based on the evaluation of the distance computed between interestingness measures. We use two techniques: agglomerative hierarchical clustering (AHC) and partitioning around medoids (PAM) to help the user graphically evaluates the behavior of interestingness measures.

1 Introduction

Knowledge quality has become an important issue of recent researches in KDD. The problem of selecting the best knowledge for a given dataset has attracted many authors in the literature. Our approach is based on the knowledge representation in the form of association rules [2], one of the few models dedicated to unsupervised discovery of rules tendencies in data. With association rules, many authors have proposed a lot of interestingness measures to evaluate the best matched knowledge from a ruleset: to select the best measures or the best rules. According to Freitas [5], two kinds of interestingness measures existing can be differentiate: objective and subjective. Subjective measures are strongly influenced by the user's goals and his/her knowledge or beliefs, and are combined to specific supervised algorithms in order to compare the extracted rules with what the user knows or wants [13] [12], rule novelty and unexpectedness in point of view of the user are captured. Objective measures are statistical indexes that depend strictly on the data structures. The definitions and properties of many objective measures are proposed and surveyed [3] [8] [16] to study the behavior of the objective measures to design a suitable measure or to help the user to select the best ones with their preferences. We focus on objective measures (called measure for short) as a natural way to discover different hidden aspects in the data.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 330–337, 2005.

Many interesting surveys on objective measures can be found in the literature. They mainly address two related research issues: the definition of the set of principles or properties that lead to the design of a good measure; their comparison from a data-analysis point of view to study measure behavior in order to help the user to select the best ones [8][16][17][10].

In this paper, we propose a new approach to evaluate the behavior of 35 interestingness measures discussed in the literature to discover the clusters of interestingness measures existing in the user's dataset. Our approach is based on the distance computed between interestingness measures by using the two clustering methods agglomerative hierarchical clustering (AHC) and partitioning around medoids (PAM) to help the user to discover the behavior of the interestingness measures studied in his/her dataset graphically.

The paper is organized as follows. In Section 2, we present the correlation and the distance between measures. In Section 3, we introduce two views for evaluating the behavior of a set of 35 measures on a dataset. Finally, we conclude and introduce some future researches.

2 Distance Between Measures

Based on the idea of measuring the the statistical surprisingness of implication theory [7] that we have mentioned in [10], we continue to extend the principles discussed from [10]. Let $R(D) = \{r_1, r_2, ..., r_p\}$ denote input data as a set of p association rules derived from a dataset D. Each rule $a \Rightarrow b$ is described by its itemsets (a, b) and its cardinalities $(n, n_a, n_b, n_{a\bar{b}})$. Let M be the set of q available measures for our analysis $M = \{m_1, m_2, ..., m_q\}$. Each measure is a numerical function on rule cardinalities: $m(a \Rightarrow b) = f(n, n_a, n_b, n_{a\bar{b}})$.

For each measure $m_i \in M$, we can construct a vector
$m_i(R) = \{m_{i1}, m_{i2}, ..., m_{ip}\}$, $i = 1..q$, where m_{ij} corresponds to the calculated value of the measure m_i for a given rule r_j.

The correlation value between any two measures $m_i, m_j\{i, j = 1..q\}$ on the set of rules R will be calculated by using a Pearson's correlation coefficient CC [15], where $\overline{m_i}, \overline{m_j}$ are the average calculated values of vector $m_i(R)$ and $m_j(R)$ respectively.

$$CC(m_i, m_j) = \frac{\sum_{k=1}^{p}[(m_{ik} - \overline{m_i})(m_{jk} - \overline{m_j})]}{\sqrt{[\sum_{k=1}^{p}(m_{ik} - \overline{m_i})^2][\sum_{k=1}^{p}(m_{jk} - \overline{m_j})^2]}}$$

In order to interpret the correlation value, we introduce the two following definitions:

Definition 2. *Correlated measures* (τ-correlation). Two measures m_i and m_j are correlated with respect to the dataset D if their absolute correlation value is greater than or equal to a threshold τ: $|CC(m_i, m_j)| \geq \tau$.

Definition 3. *Distance.* The distance d between two measures m_i,m_j is defined by:

$$d(m_i, m_j) = 1 - |CC(m_i, m_j)|$$

As both correlation and distance are symmetrical, the $q(q-1)/2$ values can be stored in one half of a table $q \times q$. We then use the distances computed from this table for both the AHC and PAM methods.

3 Measure Behavior

3.1 Data Description and Used Measures

To study the measure behavior, we try to evaluate the effect of measures based on the distance calculations for the dataset D_1. We use the categorical dataset *mushroom* (D_1) from Irvine machine-learning database repository [4]. We then generate the set of association rules (ruleset) R_1 from the dataset D_1 using the the algorithm Apriori [1]. We use 35 interestingness measures to this study (34 measures are referenced in [10] and a measure $II = 1 - \sum_{k=max(0,n_a-n_b)}^{n_{a\overline{b}}} \frac{C_{n_b}^{n_a-k} C_{n_{\overline{b}}}^{k}}{C_n^{n_a}}$). A remark is that EII[$\alpha = 1$] and EII[$\alpha = 2$] are two entropic versions of the II measure). Hereafter, we use this ruleset as our knowledge data for analysis.

Table 1. Ruleset description

Dataset	Items	Transactions	Average length of transactions	Number of rules (support threshold)	Ruleset
D_1	118	8416	22	123228 (12%)	R_1

Our aim is to discover the behavior of the measures via two views: the strong relation and the relative distance between measures occur when they are applied to the distance matrix (or distance table) calculated from R_1 (see Sec. 2). This result is useful because we can capture the different aspect or *the nature of the available knowledge* existing in the rulesets. We use the two techniques AHC and PAM for each of these views respectively.

3.2 With AHC

Fig. 1 illustrates the result computed from R_1. The horizontal line goes throughout the cluster dendrogram has the small distance 0.15 determining the clusters of measures having strong relation (strongly correlated). The assignment $\tau = 0.85 = 1 - 0.15$ of τ-correlation is used because this value is widely acceptable in the literature. The clusters are represented in details in Tab. 2.

Intuitively, the user can choose the biggest cluster in Tab. 2 contains the measures Lift, Rule Interest, Phi-Coefficient, Kappa, Similarity Index, Putative Causal Dependency, Dependency, Klosgen, Pavillon for their first choice. In this cluster we can easily see two strong related clusters with four measures for each. This cluster gives the strongest effect on evaluation the similarity between two parts of an association rule. Another observation illustrates the existence of a

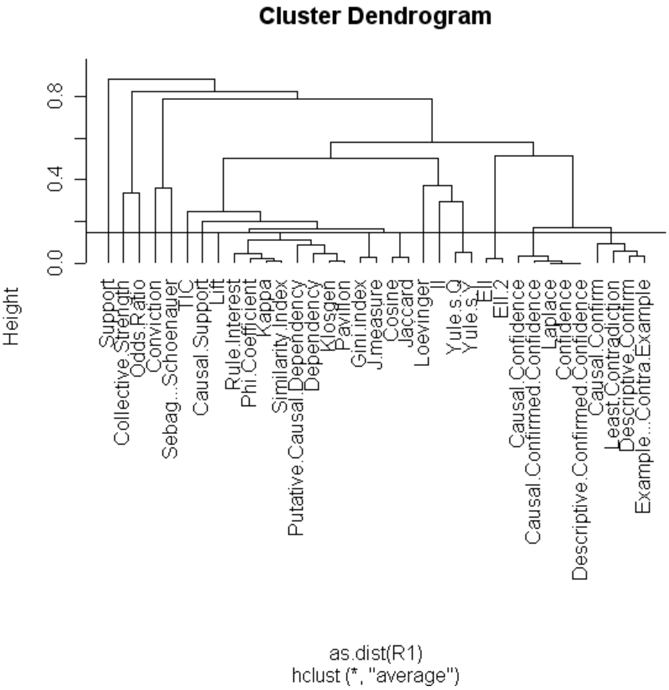


Fig. 1. View on the strong relation between measures

Table 2. Clusters of measures with AHC (distance = 0.15)

Cluster	R_1
1	Causal Confidence, Causal Confirmed-Confidence, Confidence, Descriptive Confirmed-Confidence, Laplace
2	Causal Confirm, Descriptive Confirm, Example & Contra-Example, Least Contradiction
3	Causal Support
4	Collective Strength
5	Conviction
6	Cosine, Jaccard
7	Dependency, Kappa, Klosgen, Lift, Pavillon, Phi-Coefficient, Putative Causal Dependency, Rule Interest, Similarity Index
8	EII, EII 2
9	Gini-index, J-measure
10	II
11	Loevinger
12	Odds Ratio
13	Sebag & Schoenauer
14	Support
15	TIC
16	Yule's Q, Yule's Y

confidence cluster (the first cluster in Tab. 2) with Causal Confidence, Causal Confirmed-Confidence, Laplace, Confidence, Descriptive Confirmed-Confidence. The user can then select this cluster to discover all the rules have the effect of high confidence.

This view is useful because the user can determine the strong relation between interestingness measures via the graphical representation. The hierarchical structure allows the user clearly seeing the clusters of measures that are connected closely with the hierarchical level computed.

3.3 With PAM

We can see relatively the distance between clusters by applying the principal component analysis, the number of cluster is determined from the first view (Sec. 3.2). For example, Fig. 2 illustrates the result obtained from R_1. Each symbol from Fig. 2 represents every measure in the same cluster. PAM is very useful because it gives a graphical view of clusters intuitively.

The user can now choose the aspects in the ruleset by viewing the clusters with their distances calculated (Fig. 2) based on the projection on the two principal components. The measures that have the smallest distances between them will be grouped in one cluster. In Tab. 3 the two clusters 1 (Causal Confirmed-Confidence, Descriptive Confirmed-Confidence, Confidence, Laplace, Causal Confidence) and 2 (Least Contradiction, Example & Contra-Example, Causal Confirm, Descriptive Confirm) as two different aspects the most close with the very small between-distance or separation. Then, the user can obtain automatically the representative measures for each of these two clusters are Causal Confirmed-Confidence and Example & Contra-Example. Another useful

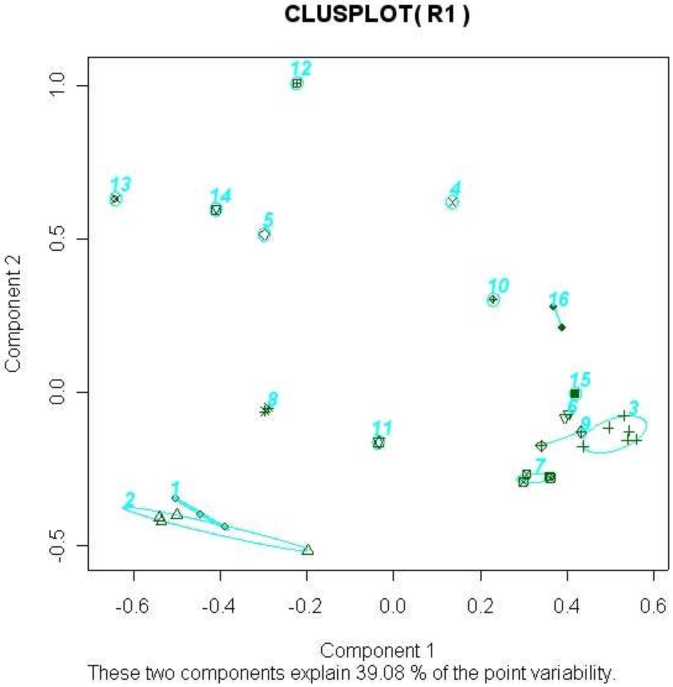


Fig. 2. Views on the relative distance between clusters of measures

Table 3. Clusters of measures with PAM

Cluster	R_1
1	Causal Confidence, Causal Confirmed-Confidence, Confidence, Descriptive Confirmed-Confidence, Laplace
2	Causal Confirm, Descriptive Confirm, Example & Contra-Example, Least Contradiction
3	Causal Support, Kappa, Lift, Phi-Coefficient, Rule Interest, Similarity Index
4	Collective Strength
5	Conviction
6	Cosine, Jaccard
7	Dependency, Klosgen, Pavillon,Putative Causal Dependency
8	EII, EII 2
9	Gini-index, J-measure
10	II
11	Loevinger
12	Odds Ratio
13	Sebag & Schoenauer
14	Support
15	TIC
16	Yule's Q, Yule's Y

information is that the diameter of the cluster 1 is smaller than the cluster 2 so this observation illustrates the strongly coherent interestingness values computed from the measures in cluster 1, representing the high value of the confidence aspect. Another choice is that the user can select in Tab. 3 one aspect formed by the cluster 3 (Causal Support, Kappa, Lift, Phi-Coefficient, Rule Interest, Similarity Index) that is very far from the two clusters 1&2 but the nearest cluster with the others such as 9,7,15 (Fig. 3) and having Kappa as the representative measure for this cluster. The user can also interest in the cluster 10 (II) in Tab. 3 standing isolated with other clusters (Fig. 3).

This view based on relative distance has an important role because it allows the user to choose the aspects that he/she takes interested by regarding the scale between them. The distance between clusters will help the user to evaluate more precisely the near or far between these aspects.

3.4 Comparing with AHC and PAM

With two different evaluations based on the two views of AHC and PAM we can obtain some interesting results: cluster that seems independent from the nature of data and the selection of rules. Comparing from Tab. 2 and Tab. 3 we can easily see sixteen clusters agreed perfectly (see Tab. 4).

To understand the behavior of the measures we will examine some important clusters in Tab. 4. For example, the first cluster (Causal Confidence, Causal Confirmed-Confidence, Confidence, Descriptive Confirmed-Confidence, Laplace) has most of the measures issued from the Confidence measure. The fifth cluster (Cosine, Jaccard) has a strong relation with the fifth property proposed by Tan et al. [16]. The sixth cluster (Dependency, Klosgen, Pavillon, Putative Causal Dependency) is necessary to distinguish between the strength of the rule $a \Rightarrow b$ from $b \Rightarrow a$. The seventh cluster (EII, EII 2) are two measures obtained with different parameters of the same original formula and very useful in evaluating the entropy of implication intensity. The ninth cluster (II) has only one measure provides the strong evaluation on the intensity of implication. The tenth cluster (Kappa, Lift, Phi-Coefficient, Rule Interest, Similarity Index) mainly gathers the

Table 4. Clusters agreed with both AHC and PAM

Cluster	R_1
1	Causal Confidence, Causal Confirmed-Confidence, Confidence, Descriptive Confirmed-Confidence, Laplace
2	Causal Confirm, Descriptive Confirm, Example & Contra-Example, Least Contradiction
3	Collective Strength
4	Conviction
5	Cosine, Jaccard
6	Dependency, Klosgen, Pavillon,Putative Causal Dependency
7	EII, EII 2
8	Gini-index, J-measure
9	II
10	Kappa, Lift, Phi-Coefficient, Rule Interest, Similarity Index
11	Loevinger
12	Odds Ratio
13	Sebag & Schoenauer
14	Support
15	TIC
16	Yule's Q, Yule's Y

measures from different properties such as symmetry, anti-symmetry [16]. The last cluster (Yule's Y, Yule's Q) gives a trivial observation because the measures are all derived from Odds Ratio measure, that is similar to the second property proposed by Tan et al. [16].

4 Conclusion

To understand the behavior of the interestingness measures on a specific dataset, we have studied and compared the various interestingness measures described in the literature to find the different aspects existing in a dataset. Our approach is the first step towards the process of evaluating the knowledge issued in the form of association rules in the domain of knowledge quality research. We use a data analysis approach based on the distance computed between interestingness measures (with two clustering methods AHC and PAM) in order to evaluate the behavior of 35 interestingness measures. These two graphically clustering methods can be used to help a user in selecting the best measures. We also determine sixteen clusters with some interesting results: cluster that seems independent from the nature of data and the selection of rules. We also evaluate the behavior of the measures on some important clusters agreed with both AHC and PAM. With this result, the decision-maker will decide what measures are interesting to capture the best knowledge.

Our future research will be investigated in introducing a new approach to facilitate the the user's decision making from the best interestingness measures to select the best association rules (the best knowledge discovered).

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. Proc. of the 20th VLDB. Santiago, Chile (1994) 487–499
2. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. Proc. of the ACM-SIGMOD Int. Conf. on Management of Data. Washington DC, USA (1993) 207–216

3. Bayardo, Jr.R.J., Agrawal, R.: Mining the most interestingness rules. KDD'1999. San Diego, CA, USA (1999) 145–154
4. Blake, C.L., Merz, C.J.: {UCI} Repository of machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html. University of California, Irvine, Dept. of Information and Computer Sciences, (1998).
5. Freitas, A.A.: On rule interestingness measures. Knowledge-Based Systems, 12(5-6). (1999) 309–315
6. Gras, R., Couturier, R., Blanchard, J., Briand, H., Kuntz, P., Peter, P.: Quelques critères pour une mesure de qualité de règles d'association. Mesures de Qualité pour la Fouille de Données, RNTI-E-1. Cépaduès Editions (2004) 3–31
7. Gras, R.: L'implication statistique - Nouvelle méthode exploratoire de données. La pensée sauvage édition (1996)
8. Hilderman, R.J., Hamilton, H.J.: Knowledge Discovery and Measures of Interestingness. Kluwer Academic Publishers (2001)
9. Huynh, X.H., Guillet, F., Briand, H.: ARQAT: an exploratory analysis tool for interestingness measures. ASMDA'05. (2005) 334–344
10. Huynh, X.H., Guillet, F., Briand, H.: Clustering interestingness measures with positive correlation. ACM ICEIS'05. (2005) 248–253
11. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York, (1990)
12. Liu, B., Hsu, W., Mun, L., Lee, H.: Finding interestingness patterns using user expectations. IEEE Trans. on Knowledge and Data Mining (11). (1999) 817–832
13. Padmanabhan, B., Tuzhilin, A. : A belief-driven method for discovering unexpected patterns. KDD'1998. (1998) 94–100
14. Piatetsky-Shapiro, G.: Discovery, analysis and presentation of strong rules. Knowledge Discovery in Databases, G. Piatetsky-Shapiro and W. Frawley editors. MIT Press, Cambridge, MA (1991) 229–248
15. Saporta, G.: Probabilité, analyse des données et statistique. Edition Technip, (1990)
16. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right objective measure for association analysis. Information Systems 29(4). (2004) 293–313
17. Vaillant, B., Lenca, P., Lallich, S.: A clustering of interestingness measures. DS'04. (2004) 290-297

Unit Volume Based Distributed Clustering Using Probabilistic Mixture Model*

Keunjoon Lee[1], Jinu Joo[2], Jihoon Yang[2], and Sungyong Park[2]

[1] Kookmin Bank, 27-2,
Yeouido-Dong, Yeoungdeungpo-Ku, Seoul, Korea
leekjsg@hanmail.net
[2] Department of Computer Science and
Interdisciplinary Program of Integrated Biotechnology, Sogang University,
1 Shinsoo-Dong, Mapo-Ku, Seoul 121-742, Korea
jujoo@mllab.sogang.ac.kr,
{yangjh, parksy}@sogang.ac.kr

Abstract. Extracting useful knowledge from numerous distributed data repositories can be a very hard task when such data cannot be directly centralized or unified as a single file or database. This paper suggests practical distributed clustering algorithms without accessing the raw data to overcome the inefficiency of centralized data clustering methods. The aim of this research is to generate unit volume based probabilistic mixture model from local clustering results without moving original data. It has been shown that our method is appropriate for distributed clustering when real data cannot be accessed or centralized.

1 Introduction

Data clustering is a method of grouping or partitioning similar patterns to subsets. Patterns that are grouped in the same cluster can be analyzed to have closer relationship than other patterns in different clusters. Clustering algorithms are applied in various areas such as visualization, pattern recognition, learning theory, computer graphics, and neural networks [1]. Recently, issues on distributed clustering has arisen. Distributed clustering is particularly useful when distributed data are hard to be centralized because of privacy, communication cost, and the limit of storage. Against this background, we suggest practical distributed algorithms.

Our method runs in three steps. First, local clustering results are gathered without moving the original data. Second, based on the received results (mean and covariance that represent the local clusters) each unit volume in the global data space is assigned to clusters with the highest probability. Finally, clusters with similar probability distributions are merged.

* This work is supported by grant No. R01-2004-000-10689-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 338–345, 2005.

Unit volume based distributed clustering has three advantages compared to other existing distributed clustering methods. First, distributed data are clustered in parallel without physically moving the data, and each local site may run different clustering algorithms. Second, the size of the unit volume can be configured freely based on the size and domain of the data, so that global clustering can approximate on local data distributions. Third, when merging normally distributed models in global clustering, a diversity of mixture models can be deduced during the process which will lead to near optimal clustering models that represent overall data.

Two types of distributed data exist: *horizontally* distributed data where data are distributed by instance, and *vertically* distributed data where data are distributed by attributes. In this paper we concern only horizontally distributed data to perform clustering.

2 Related Work

Two major types of distributed algorithms exist depending on the relationship between global clustering and local clustering. The first type is where the local clustering algorithm is related to the global clustering algorithm. The other type is where the local and global clustering is independent and does not interfere with each other. The former includes DBDC [2] and k-windows distributed clustering [3,4], and the latter includes privacy preserving distributed clustering [5].

DBDC is an extension of DBSCAN [6] which is a density based clustering algorithm. It performs DBSCAN in each local clustering phase and DBSCAN is used once again in global clustering. This method requires large memory space for saving every distance value between data as a tree. And global clustering algorithm is deeply related to local clustering algorithms. Therefore, different clustering algorithms can not be introduced separately for local clustering and global clustering.

K-windows distributed clustering uses k-windows algorithm to cluster data at each local site. Global clustering is performed by merging local clusters of high similarity. The drawback is that there are too many parameters that the user must define, and local clustering algorithm is bound to k-windows clustering algorithm, which means that the user doesn't have any freedom to use other clustering algorithms for local clustering.

Privacy preserving distributed clustering algorithm complements the inherent drawbacks of DBDC and k-windows distributed clustering, and let local clustering algorithms be independent from global clustering algorithms. In other words, each local site is allowed to use different clustering algorithms to cluster its local data. The drawback is that the user must define the universal set of potential clustering models and produce a dataset through MCMC sampling before global clustering.

Based on these backgrounds about distributed clustering, our algorithm not only allows each local site to choose its clustering algorithm freely but the user only needs to decide the unit volume size which gives better chances to deduce various mixture models.

3 Unit Volume Based Distributed Clustering

3.1 Local Clustering

If we let random variable $\mathbf{X} = (X_1, X_2, ..., X_n)$ represent features of the instance, a probability density function indicates a local cluster from the local clustering results. That is, instances in the same local cluster are represented by the mean and the covariance values. Then the clustering results (i.e. mean and covariance value of each local cluster) are sent to the global site to be used in global clustering. Therefore, different clustering algorithms are able to run in each local site. The overall conceptual diagram is shown in Fig 1.

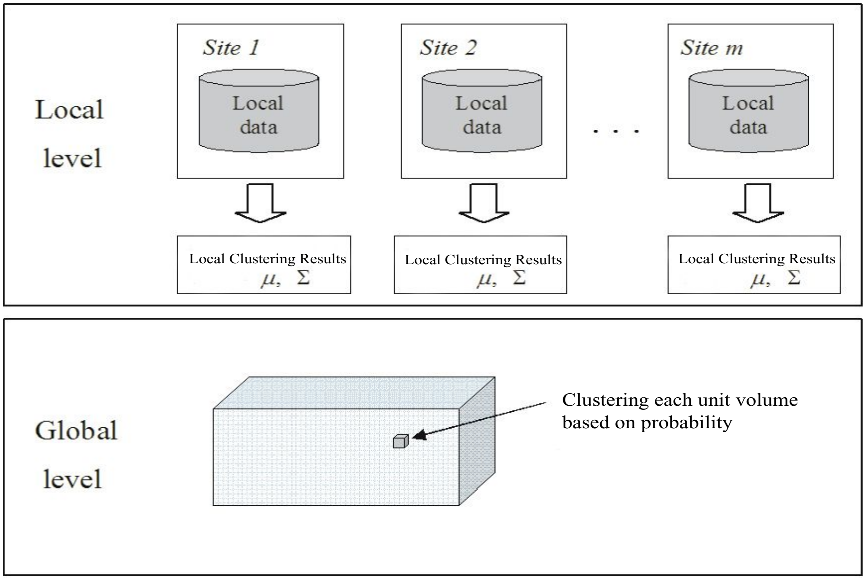

Local level
Site 1
Local data
Site 2
Local data
. . .
Site m
Local data
Local Clustering Results μ, Σ
Local Clustering Results μ, Σ
Local Clustering Results μ, Σ
Global level
Clustering each unit volume based on probability

Fig. 1. Conceptual diagram of unit volume based distributed clustering

3.2 Global Clustering

In global clustering, distribution of instances are presumed by the results of local clustering. First, data space is divided into unit volume $V = h^n$ equally where $n \in \Re$ is the number of features and h is the given unit length. Next, $center_V$, which is the central point of each unit volume, is used to decide which cluster the unit volume is most likely to be assigned to, based on the probability density function. The probability density function is calculated by the mean and the variance calculated by each local clustering. Therefore each unit volume is assigned to clusters depending on the results of,

$$\begin{aligned} Cluster(center_V) &= argmax_{c_i \in C} \int_V f(X = center_V) \\ &= argmax_{c_i \in C}\ volume_V \\ &\times \frac{1}{\sigma\sqrt{2\pi}} \exp[-\frac{1}{2}(center_V - \mu_{c_i})^t \Sigma_{c_i}^{-1}(center_V - \mu_{c_i})] \end{aligned} \tag{1}$$

where c_i is the i_{th} cluster, C is the set of clusters, $volume_V$ is the volume of one unit, μ_{c_i}, Σ_{c_i}, σ are mean, variance, standard deviation values of c_i, respectively.

3.3 Merging

The merging process is to reflect the similarity between local clusters. Two methods, mean based merging and unit volume based merging, have been introduced. We present the following notations to facilitate the description on merging clusters:

- $C^{Global} = \{c_1, c_2, ..., c_i\}$ is a set of global clusters.
- $V_{total} = \{V_1, V_2, ..., V_N\}$ is a set of unit volumes in the data space.
- μ_{c_i} is the mean point of cluster c_i.
- $|D_i|$ is number of instances in cluster c_i.
- $p_{c_i}(x)$ is the approximated integration of probability density function f_{c_i} with unit volume where central point is x.
- p_{merge} is the probability of sampling the sample point from mixture model of two merged clusters.
- p_{split} is the probability of sampling the sample point from two separate clusters.
- $mixtureModel(f_{c_i}, f_{c_j})$ is a mixture of two normally distributed models, f_{c_i} and f_{c_j}, with different weights [7].

If $p_{merge} > p_{split}$ then the two clusters are merged and $mixtureModel(f_{c_i}, f_{c_j})$ is calculated with weights given by ratio of $|D_i|$. The overall algorithm is:

```
Function $mergingGlobalClusters$ $(C^{Global}, V_{total}, mergeType)$
    if ($mergeType$==meanBased) then
      for $i = 1$ to $|C^{Global}|$ do
          $\mu_{c_i} = E[center_V]\ \forall center_V \in c_i$
      endfor

    for all adjacent $c_i, c_j \in C^{global}$ do
       $w_i = \frac{|D_i|}{|D_i|+|D_j|}, w_j = \frac{|D_j|}{|D_i|+|D_j|}$
       calculate $p_{merge}$, $p_{split}$
```

```
            if $p_{merge} > p_{split}$ then
              $f_{c_i} = mixtureModel(f_{c_i}, f_{c_j})$
              $c_i \leftarrow \{c_i \bigcup c_j\}$ /* merging */
            endif
        endfor
    **Endfunction**
```

Mean Based Merging. Mean points in each cluster are considered for merging. Therefore,

$$p_{merge} = w_i p_{c_i}(\mu_{c_i}) \times w_j p_{c_j}(\mu_{c_j}) \tag{2}$$

$$p_{split} = p_{c_i}(\mu_{c_i}) \times p_{c_j}(\mu_{c_j}) \tag{3}$$

Unit Volume Based Merging. Every central point of the unit volume in the cluster is considered for merging. Therefore,

$$p_{merge} = \prod_{center_V \in c_i, c_j} [w_i p_{c_i}(center_V) + w_j p_{c_j}(center_V)] \tag{4}$$

$$p_{split} = [\prod_{center_V \in c_i} p_{c_i}(center_V)] \times [\prod_{center_V \in c_j} p_{c_j}(center_V)] \tag{5}$$

4 Experiments

4.1 Description of Datasets

Five different datasets were tested in the experiment. A real-world dataset, Iris, was from the UCI Machine Learning Repository [1]. The other four datasets, 2D3C3S, 2D3C3Sbiased, 3D3C3S, 3D3C3Sbiased (D: number of attributes, C: number of class, S: number of distributed sites, biased: dataset that has specific class biased in a certain site) were artificially generated. For example, 2D3C3S is generated by selecting 3000 instances randomly out of three different type of normally distributed models each corresponding to three different type of classes, while 3D3C3S is generated by selecting 1800 instances randomly from three different type of normally distributed models. Table 1 shows the number of features, classes, clusters and instances of the datasets used in our experiments.

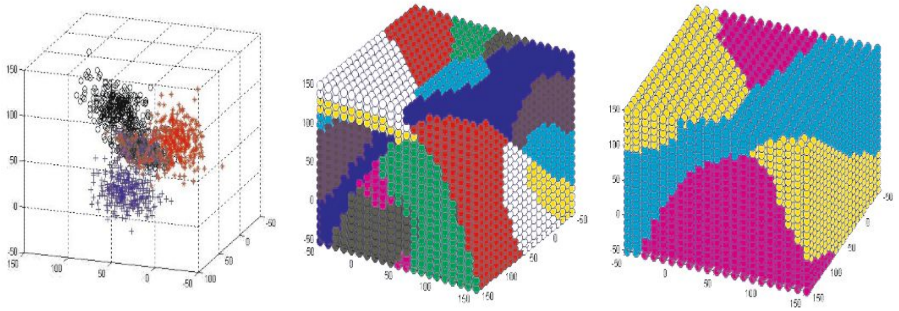
4.2 Experimental Results

In this experiment, k-means clustering has been adopted for local clustering and the classification accuracy was used to measure the quality of clustering. The accuracy was calculated by comparing instances in resulting clusters with actual classes. In other words, accuracy is the hit ratio of the instances that matches with the real class which is specified in formula (6).

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html

Table 1. Number of features, classes, clusters and instances of each dataset

Dataset	Feature	Class	Cluster	Instance
Iris	4	3	3	150
2D3C3S	2	3	3	3000
2D3C3S(biased)	2	3	3	300
3D3C3S	3	3	3	1800
3D3C3S(biased)	3	3	3	900



(a) True Model

(b) Global Clustered Model

(c) Merged Model

Fig. 2. Representation of 3D3C3S dataset

$$accuracy_{Cluster} = \frac{number\ of\ hit\ instance}{number\ of\ total\ instance} \tag{6}$$

In Fig.2 we can see the clustered models with 3D3C3S data. Fig.2(a) is the true model of 3D3C3S data, representing each data point in the three dimensional feature space. Fig.2(b) shows the results of global clustering. Fig.2(c) is the final model after merging. By comparing Fig.2(a) with Fig.2(c) we can see that our clustered model approximates the true model accurately.

Table 2 shows the accuracies for the five datasets compared among the four different algorithms. **Global K-means** indicates experiments performed with

Table 2. Comparison of different clustering algorithms using global and local k-means algorithm and unit volume based distributed clustering with mixture models

	Global	**Local**	**MeanBased**	**VolumeBased**
Dataset	**K-means**	**K-means(avg)**	**GC**	**GC**
Iris	86.2%	84.6%	73.5%	79.3%
2D3C3S	84.2%	83.1%	76.8%	78.2%
2D3C3S(biased)	81.3%	52.6%	62.4%	69.3%
3D3C3S	82.1%	79.8%	71.4%	70.6%
3D3C3S(biased)	77.0%	49.5%	59.7%	60.4%

all instances running k-means algorithm 10 times. **Local K-means(avg)** indicates average accuracy calculated by doing k-means on each local site. **Mean-Based GC** is our unit volume algorithm performed by merging clusters with mean points while **Volume Based GC** indicates merging considering all unit volume's center points. From the results, **Global K-means** showed the highest accuracies because clustering was performed on the whole dataset. Note that **Local K-means(avg)** produced better results than both **MeanBased GC** and **VolumeBased GC** for unbiased data, but the latter algorithms outperformed the former algorithm for biased data. This is because **Local K-means** simply computes the averaged results regardless of the distribution of instances, while **MeanBased GC** and **VolumeBased GC** generate appropriate clusters considering all the instances at every site. Additionally as **MeanBased GC** and **VolumeBased GC** scarcely have significant difference on accuracy. **Mean-Based GC** is recommended when dealing with large datasets due to the low computational overhead.

From this experiment we can say that the unit volume distributed clustering performed well, considering real world data being biased in distributed environments, and gives the merit that distributed data can be clustered without being directly accessed or physically moved from one site to another with high accuracy. Among the unit volume distributed clustering algorithms, **MeanBased GC** turned out to be our gold standard algorithm due to low computational cost and high accuracy.

5 Summary and Discussion

Throughout this paper we have proposed a method that clusters global data probabilistically based on the unit volume without physically moving or directly accessing distributed data. Likewise, similar clusters are merged considering the mean points or probability of unit volume's central points in mixture models at global clustering stage. The method introduced in this paper proved to show better performance when distributed data is impossible to reach directly and instance classes are biased at certain sites, in particular.

Some of future research directions include: First, setting the definition of the unit volume to describe clusters more naturally; Second, further experiments with various types of distributed data; Third, using other measures of cluster similarity such as the distance between cluster centers; Finally, improving our global clustering method to overcome its limited capability and to handle data in high dimensional space.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley and Sons Inc. (2000)
2. Januzaj, E., Kriegel, H.P., Pfeifle, M.: Towards effective and efficient distributed clustering. In: International Workshop on Clustering Large Data Sets (ICDM). (2003)

3. Vrahatis, M.N., Boutsinas, B., Alevizos, P., Pavlides, G.: The new k-windows algorithm for improving the k-means clustering algorithm. Journal of Complexity **18** (2002) 375–391
4. Tasoulis, D.K., Vrahatis, M.N.: Unsupervised distributed clustering. In: The IASTED International Conference on Parallel and Distributed Computing and Networks, as part of the Twenty-Second IASTED International Multi-Conference on Applied Informatics, Innsbruck, Austria (2004)
5. Merugu, S., Ghosh, J.: Privacy-preserving distributed clustering using generative models. In: The Third IEEE International Conference on Data Mining (ICDM'03). (2003)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In Simoudis, E., Han, J., Fayyad, U., eds.: Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, AAAI Press (1996) 226–231
7. Trivedi, K.S.: Probability and statistics with reliability, queuing and computer science applications. John Wiley and Sons Inc. (2002)

Finding Significant Web Pages with Lower Ranks by Pseudo-Clique Search

Yoshiaki Okubo, Makoto Haraguchi, and Bin Shi

Division of Computer Science,
Graduate School of Information Science and Technology,
Hokkaido University,
N-14 W-9, Sapporo 060-0814, Japan
{yoshiaki, mh}@ist.hokudai.ac.jp

Abstract. In this paper, we discuss a method of finding useful clusters of web pages which are significant in the sense that their contents are similar or closely related to ones of higher-ranked pages. Since we are usually careless of pages with lower ranks, they are unconditionally discarded even if their contents are similar to some pages with high ranks. We try to extract such hidden pages together with significant higher-ranked pages as a cluster.

In order to obtain such clusters, we first extract semantic correlations among terms by applying *Singular Value Decomposition*(SVD) to the term-document matrix generated from a corpus w.r.t. a specific topic. Based on the correlations, we can evaluate potential similarities among web pages from which we try to obtain clusters. The set of web pages is represented as a weighted graph G based on the similarities and their ranks. Our clusters can be found as *pseudo-cliques* in G. We present an algorithm for finding Top-N weighted pseudo-cliques. Our experimental result shows that quite valuable clusters can be actually extracted according to our method.

1 Introduction

We often try to obtain useful information or knowledge from web pages on the Internet. *Information retrieval* (IR) *systems* are quite powerful and helpful tools for this task. For instance, *Google* is well known as a popular IR system with a useful search engine. Given some keywords we are interested in, such a system shows a list of web pages that are related to the keywords. These pages are usually ordered by some ranking mechanism adopted in the system. For example, the method of *PageRank* [1] adopted in Google is widely known to provide a good ranking.

In general, only some of the higher-ranked pages are actually browsed and the others are discarded as less important ones, since the list given by the system contains a large number of pages. However, such a system presents just one candidate of ranking from some viewpoint. Therefore, there might exist many pages which are unfortunately lower-ranked but are significant for us. More concretely

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 346–353, 2005.

speaking, the ranking by PageRank is determined based on the link structure of each web page. For example, pages without enough links from others tend to be lower-ranked even if they have significant contents similar to higher ranked pages. From this point of view, it would be worth investigating a framework in which such implicitly significant pages are listed together with higher-ranked pages. We discuss in this paper a method for finding useful clusters of web pages which are significant in the sense that their contents are similar or closely related to ones of higher-ranked pages.

1.1 Similarities Among Web Pages

In order to realize it, we first extract semantic correlations among terms by applying *Singular Value Decomposition*(SVD) [3] to the term-document matrix generated from a corpus gathered with respect to a specific topic. Given a set of ranked web pages for which we try to extract clusters, we can evaluate potential similarities among them based on the semantic correlations of terms. In previous approaches, similarities among web pages are often determined based on the link structure of web pages [2]. More concretely speaking, it has been considered that web pages with similar topical contents have dense links among them. Such a link structure might roughly reflect similarities among relatively *mature* pages. However, many interesting pages are newly released day by day and it is often difficult to expect a dense link structure of fresh pages. As the result, based on the link-based approach, we will fail in finding similarities among such new pages even if they have similar contents. On the other hand, we try to capture similarities among web pages independently of their link structure.

1.2 Extracting Clusters by Clique Search

The set of web pages is then represented as a weighted undirected graph based on the similarities and their ranks. If a pair of web pages has a similarity higher than a given threshold, they are connected by an edge. Moreover, each vertex (i.e. a web page) is assigned a weight so that higher-ranked pages have higher weights. Our clusters can be extracted by finding *pseudo-cliques* in the graph G. A pseudo-clique is defined as the union of several maximal cliques in G with a required degree of overlap. Simple theoretical properties of pseudo-cliques are presented. Based on the properties, we can obtain some pruning rules for pseudo-clique search. We design a depth-first algorithm for finding pseudo-cliques whose weights (evaluation values) are in the top N. Our preliminary experimental result shows that a quite valuable cluster can be actually extracted as a pseudo-clique in G.

One might claim that a naive method would be sufficient for extracting clusters consisting of similar higher-ranked and lower-ranked pages. That is, for each web page with a higher rank, we can gather lower-ranked pages similar to the higher-ranked one. As well as this kind of clusters, our method can extract other various kinds of clusters simultaneously by changing the weighting of web pages in our graph construction process. Under some weighting, for example, a cluster consisting of several pages which are moderately ranked might be obtained as in the top N. In this sense, our method includes such a naive method.

Our method for extracting clusters by clique search is a general framework. The literature [6,9] has investigated methods for finding appropriate *data abstractions* (groupings) of attribute values for classification problems, where each abstraction is extracted as a weighted exact clique. A gene expression data has been also processed in [7]. A cluster consisting of genes which behave similarly is extracted as an exact clique. The current pseudo-clique search can be viewed as an extension of these previous search methods for exact cliques [6,7,8,9].

Our clique search-based method has advantage over previous clustering methods in the following points. In the traditional *hierarchical* or *partitional* clustering, the whole set of data is divided into some clusters. Although the number of clusters is usually controlled by a user-defined parameter, it is well known that providing an adequate value for the parameter is not so easy. Under an inadequate parameter setting, we often obtain many useless clusters. From the computational point of view, the cost for producing such useless clusters will be quite waste. On the other hand, in our method, we can extract *only* nice clusters whose evaluation values are in the top-N, where N can be given arbitrarily. In this sense, we will never suffer from quite useless clusters. Furthermore, extracting only nice clusters has an advantage in the computation. We can enjoy a branch-and-bound search in order to extract them. In our search, we do not have to examine many branches concerning clusters not in the top N.

2 Semantic Similarity Among Web Pages

In order to find clusters of web pages, we have to measure similarities among web pages. For the task, we follow a technique in *Information Retrieval*(IR) [3].

2.1 Term-Document Matrix

Let $\mathcal{D}$ be a set of documents and $\mathcal{T}$ the set of terms appeared in $\mathcal{D}$ [1]. We first remove too frequent and too infrequent terms from $\mathcal{T}$. The set of remaining terms, called *feature terms*, is denoted by $\mathcal{T}^*$. Supposing $|\mathcal{T}^*| = n$, each document $d_i \in \mathcal{D}$ can be represented as an n-dimensional document vector $\boldsymbol{d_i} = (tf_{i1}, \ldots, tf_{in})^T$, where tf_{ij} is the frequency of the term $t_j \in \mathcal{T}^*$ in the document d_i. Thus, $\mathcal{D}$ can be translated into a *term-document matrix* $(\boldsymbol{d_1}, \ldots, \boldsymbol{d_{|\mathcal{D}|}})$.

2.2 Extracting Semantic Similarity with SVD

For the term-document matrix, we apply *Singular Value Decomposition*(SVD) in order to extract correlations among feature terms [3].

An $m \times n$ matrix A can be decomposed by applying SVD as $A = U\Sigma V^T$, where U and V are $m \times m$ and $n \times n$ orthogonal matrices, respectively. Each

[1] In order to obtain such terms from documents without spaces among words (like Japanese documents), we need to apply *Morphological Analysis* to $\mathcal{D}$.

column vector in U (V) is called a left (right) singular vector. Σ is an $m \times n$ matrix of the form

$$\Sigma = \left[\begin{array}{ccc|c} \sigma_1 & & O & \\ & \ddots & & O_{r\times(n-r)} \\ O & & \sigma_r & \\ \hline & O_{(m-r)\times r} & & O_{(m-r)\times(n-r)} \end{array}\right],$$

where $rank(A) = r$ ($r \leq \min\{m, n\}$) and σ_i is called a *singular value*. First r left singular vectors $\boldsymbol{u_1}, \ldots, \boldsymbol{u_r}$ correspond to a orthonormal basis and define a new subspace of the original one in which column vectors of A exist, where the $m \times r$ matrix $(\boldsymbol{u_1}, \ldots, \boldsymbol{u_r})$ is denoted by U_r.

Let us assume the matrix A is a term-document matrix generated from a set of documents. Intuitively speaking, by applying SVD to A, we can capture *potential but not presently evident* correlations among the terms. Highly semantically correlated terms give a base vector $\boldsymbol{u_i}$ and define a dimension corresponding to a compound term. Such new base vectors define a new subspace based on compound terms. For documents d_1 and d_2 not in A, therefore, if they are projected on the subspace, we can find similarity between them based on the semantic correlations among terms captured from the original documents in A.

In order to take such semantic similarities of web pages into account, we prepare a *corpus* of documents written about some specific topic. Then by applying SVD to the term-document matrix generated from the corpus, we obtain a subspace reflecting semantic correlations among terms in the corpus. Let U_r be the orthonormal basis defining the subspace [2].

Besides the corpus, with some keywords related to the corpus topic, we retrieve a set of web pages $\mathcal{P}$ from which we try to obtain clusters. Using the same feature terms for the corpus, each document $p_i \in \mathcal{P}$ is represented as a vector $\boldsymbol{p_i} = (tf_{i1}, \ldots, tf_{in})^T$, where tf_{ij} is the frequency of the feature term t_j in p_i. Then each web page $\boldsymbol{p_i}$ is projected on the subspace as $\boldsymbol{p_i^r} = U_r^T \boldsymbol{p_i}$.

A similarity between web pages p_i and p_j, denoted by $sim(p_i, p_j)$, is defined based on the standard *cosine measure*, that is, $sim(p_i, p_j) = \frac{\boldsymbol{p_i^r} \cdot \boldsymbol{p_j^r}}{\|\boldsymbol{p_i^r}\| \times \|\boldsymbol{p_j^r}\|}$.

3 Finding Clusters by Top-*N* Pseudo-Clique Search

3.1 Graph Representation of Web Pages

Let $\mathcal{P}$ be a set of web pages from which we try to extract clusters. In order to find our clusters, $\mathcal{P}$ is represented as an undirected weighted graph G.

Assume we computed the semantic similarities among pages in $\mathcal{P}$ according to the procedure just discussed above. Let δ be a similarity threshold. Each

[2] In IR, we do not always use r left singular vectors. A part of them, that is, $U_k = (\boldsymbol{u_1}, \ldots, \boldsymbol{u_k})$ ($k < r$) is usually used for *approximation*. Such an approximation with U_k is called *Latent Semantic Indexing* (LSI) [3].

page $p_i \in \mathcal{P}$ corresponds to a vertex in G. For any web pages $p_i, p_j \in \mathcal{P}$, if $sim(p_i, p_j) \geq \delta$, then they are connected by an edge. Furthermore, we assign a weight to each vertex (page) based on its rank, where a higher-ranked page is assigned a larger weight. The weight of a page p is referred to as $w(p)$.

3.2 Top-*N* Weighted Pseudo-Clique Problem

Our cluster of similar pages can be obtained as a weighted *pseudo-clique* in the graph G. In fact, we obtain only nice clusters by extracting maximal weighted pseudo-cliques whose evaluation values are in the top-N. Before giving the problem description, we first define *degree of overlap* for a class of maximal cliques.

Definition 1 (Degree of Overlap for Maximal Clique Class). Let $\mathcal{C} = \{C_1, \ldots, C_m\}$ be a class of maximal cliques. The *degree of overlap* for $\mathcal{C}$, denoted by $overlap(\mathcal{C})$, is defined as $overlap(\mathcal{C}) = \min_{C_i \in \mathcal{C}} \left\{ \left| \cap_{C_j \in \mathcal{C}} C_j \right| / |C_i| \right\}$. ■

Using the notion of overlap degree, our pseudo-cliques is defined as follows.

Definition 2 (Pseudo-Clique). Let $\mathcal{C} = \{C_1, \ldots, C_m\}$ be a class of maximal cliques in a graph. $pseudo(\mathcal{C}) = \cup_{C_i \in \mathcal{C}} C_i$ is called a *pseudo-cliques* with the overlap degree $overlap(\mathcal{C})$. Its *size* and *weight* (evaluation value) are given by $|pseudo(\mathcal{C})|$ and $w(pseudo(\mathcal{C})) = \Sigma_{v \in pseudo(\mathcal{C})} w(v)$, respectively [3]. Moreover, the shared vertices, $\bigcap_{C_i \in \mathcal{C}} C_i$, is called the *core.* ■

We can now define the problem of finding Top-N weighted pseudo-cliques.

Definition 3 (Top-N Weighted Maximal τ Pseudo-Clique Problem). Let G be a graph and τ a threshold for overlap degree. The *Top-N Weighted Maximal τ Pseudo-Clique Problem* is to find any maximal pseudo-clique in G such that its overlap degree is greater than or equal to τ [4] and its weight is in the top N. ■

3.3 Algorithm for Finding Top-*N* Weighted Pseudo-Cliques

We present here an algorithm for finding Top-N weighted pseudo-cliques.

In our search, for a clique Q in G, we try to find a τ-valid pseudo-clique $\tilde{C}$ whose core is Q. In order to precisely discuss how it can be found, we introduce a notion of *extensible candidates* for a given clique.

Definition 4 (Extensible Candidates). Let G be a graph and Q a clique in G. A vertex $v \in V$ adjacent to any vertex in Q is called an *extensible candidate* for Q. The set of extensible candidates is denoted by $cand(Q)$. ■

From the definition, we can easily observe the followings.

[3] The weight of pseudo-clique is not restricted to the sum of vertex weights. Any monotone weight under the set inclusion can be accepted in the following discussion.

[4] Such a pseudo-clique is said to be *τ-valid.*

Observation 1. Let Q and Q' be cliques in G such that $Q \subseteq Q'$. Then, $cand(Q) \supseteq cand(Q')$ and $w(Q) + w(cand(Q)) \geq w(Q') + w(cand(Q'))$ hold, where $w(Q)$ is the weight of the clique Q. ■

Note here that the weight of a pseudo-clique with the core Q is *at most* $w(Q) + w(cand(Q))$. Therefore, a simple theoretical property can be derived.

Observation 2. Let Q be a clique. Assume we already have *tentative* Top-N weighted maximal pseudo-cliques and the minimum weight of them is w_{min}. If $w(Q) + w(cand(Q)) < w_{min}$ holds, then for any extension Q' of Q[5], there exists no pseudo-clique with the core Q' whose weight is in the top N. ■

Assume that a τ-valid pseudo-clique $\tilde{C}$ contains a clique Q as its core. $\tilde{C}$ can be obtained as the union of any maximal clique C such that $Q \subset C$ and $|Q|/|C| \geq \tau$. It should be noted here that for such a clique C, there exists a maximal clique D in $G(cand(Q))$ such that $Q \cup D = C$, where $G(cand(Q))$ is the subgraph induced by $cand(G)$. That is, finding any maximal clique D in $G(cand(Q))$ such that $|Q|/(|Q| + |D|) \geq \tau$ is sufficient to obtain the pseudo-clique $\tilde{C}$. Although one might claim that such a task is quite expensive from the computational point of view, we can enjoy a pruning in the maximal clique search based on the following observation.

Observation 3. For a clique Q in G, let us assume that we try to find a τ-valid pseudo-clique $\tilde{C}$ whose core is Q. For a clique D in $G(cand(Q))$, if $|D| > (\frac{1}{\tau} - 1) \cdot |Q|$, then any extension of D is useless for obtaining $\tilde{C}$. ■

Furthermore, in a certain case, we can immediately obtain a pseudo-clique without finding maximal cliques in $G(cand(Q))$.

Observation 4. Let Q be a clique in G and τ a threshold for overlap degree. If the followings hold, then $Q \cup cand(Q)$ is a τ-valid maximal pseudo-clique with the core Q.

- $(\frac{1}{\tau} - 1) \cdot |Q| \geq k$ holds, where k is an upper bound of the maximum clique size in $G(cand(Q))$.
- For any $v \in cand(Q)$, its degree in $G(cand(Q))$ is less than $|cand(Q)| - 1$. ■

Upper bounds for the maximum clique size have been widely utilized in efficient depth-first branch-and-bound algorithms for finding maximum cliques [4,5,9]. The literature [5] has argued that the (*vertex*) *chromatic number* χ can provide the tightest upper bound. However, identifying χ is an NP-complete problem. Therefore, approximations of χ are usually computed [4,5,9].

Based on the above properties, Top-N τ-valid weighted pseudo-cliques can be extracted with a *depth-first hybrid search.* For each core candidate Q, its surroundings are explored by finding maximal cliques in $G(cand(Q))$. In the search for core candidates, we can enjoy a pruning based on Observation 2. In the surroundings search, a pruning based on Observation 3 can be applied. Furthermore, for some core candidates, our surroundings search can be skipped based on Observation 4. More precise description of our algorithm is found in [10].

[5] For a pair of cliques Q and Q', if $Q \subset Q'$, then Q' is called an *extension* of Q.

4 Experimental Result

In this section, we present a result of our experimentation conducted on a PC with Xeon-2.4 GHz and 512MB RAM.

We have manually prepared a (Japanese) corpus with 100 documents written about “Hokkaido” and have selected 211 feature terms from the corpus. Applying SVD to the 211 × 100 matrix, a 98-dimensional subspace has been obtained.

Besides the corpus, we have retrieved 829 web pages by Google with the keywords “Hokkaido” and “Sightseeing”. The 211 × 829 term-document matrix for the pages has been projected on the subspace in order to capture semantic similarities among pages. Under the setting of $\delta = 0.95$, we have constructed a weighted graph G from the projected pages. The numbers of vertices and edges are 829 and 798, respectively. Each page d has been given a weight defined as $w(d) = 1/rank(d)^2$, where $rank(d)$ is the rank of d assigned by Google (PageRank). We have tried to extract Top-15 weighted 0.8-pseudo cliques in the graph.

Among the extracted 15 clusters (pseudo-cliques), the authors especially consider that the 11^{th} one is quite interesting. It consists of 6 pages with the ranks, 11^{th}, 381^{th}, 416^{th}, 797^{th}, 798^{th} and 826^{th}. The 11^{th} and 328^{th} pages are index pages for travel information and we can make reservations for many hotels via the pages. The 416^{th} page is an article in a private BBS site for travels. It reports on a private travel in Hokkaido and provides an actual information about hotels and enjoyable foods. The 797^{th} and 798^{th} personal pages give the names of two hotels serving smorgasbords in Hokkaido. The 826^{th} page lists hotels most frequently reserved in a famous travel site in 2004. Thus, their contents are very similar in the sense that all of them give some information about accommodations in Hokkaido, especially about hotels and foods. When we try to make travel plans for sightseeing in Hokkaido, we would often care about hotels and foods as important factors. In such a case, the cluster will be surely helpful for us.

Similar to the literature [8], we can find Top-N clusters of web pages by an *exact* clique search. In that case, however, our 11^{th} cluster can never be obtained. The cluster (that is, a pseudo-clique) consists of two exact maximal cliques: $\{11^{th}, 382^{nd}, 797^{th}, 798^{th}, 826^{th}\}$ and $\{382^{nd}, 416^{th}, 797^{th}, 798^{th}, 826^{th}\}$. In the exact case, the former is 11^{th} cluster, whereas the latter 343^{rd} one. It should be noted that the 416^{th} page will be invisible unless we specify a large N for Top-N (about 350). However, it would be impractical to specify such a large N because many clusters are undesirably extracted. Although 416^{th} page has valuable contents as mentioned above, we will lose a chance to browse it.

In case of pseudo-clique search, the 343^{rd} exact clique can be absorbed into the 11^{th} clique to form a pseudo-clique. That is, the 343^{rd} cluster can be drastically raised its rank. As the result, 416^{th} page can become visible by just specifying a reasonable N. Thus, our chance to get significant lower-ranked pages can be enhanced with the help of pseudo-cliques.

Our experimental result also shows that our pruning rules can be applied very frequently in our search. The number of cores actually examined was 69981 and our pruning were invoked at 40832 nodes of them. As the result, the total computation time was just 0.847 second.

5 Concluding Remarks

In this paper, we discussed a method of finding clusters of web pages which are significant in the sense that their contents are similar or closely related to ones of higher-ranked pages. Although we are usually careless of pages with lower ranks, they can be *explicitly* extracted together with significant higher-ranked pages. As the result, our clusters can provide new valuable information for users.

Obtained clusters are very sensitive to the assignment of vertex weights in our graph construction process. Although the reciprocal of the page rank squared currently adopted seems to be promising, we have to examine any other candidates. Furthermore, the required degree of overlap for pseudo-cliques also affects which clusters can be found. In order to obtain good heuristics for these settings, further experimentations should be conducted.

A meaningful cluster should have a clear explanation why the pages in the cluster are grouped together or what the common features in the cluster are. Our current method, unfortunately, does not have any mechanism to provide it clearly. If such a explanation mechanism is integrated, our clusters would be more convincing. An improvement on this point is currently ongoing.

References

1. L. Page, S. Brin, R. Motwani and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web”, http://dbpubs.stanford.edu/pub/1999-66, 1999.
2. A. Vakali, J. Pokorný and T. Dalamagas, “An Overview of Web Data Clustering Practices”, Proceedings of the 9th International Conference on Extending Database Technology - EDBT’04, Springer-LNCS 3268, pp. 597 - 606, 2004.
3. K. Kita, K. Tsuda and M. Shishibori, “Information Retrieval Algorithms”, Kyoritsu Shuppan, 2002 (in Japanese).
4. E. Tomita and T. Seki, “An Efficient Branch-and-Bound Algorithm for Finding a Maximum Clique”, Proceedings of the 4th International Conference on Discrete Mathematics and Theoretical Computer Science - DMTCS’03, Springer-LNCS 2731, pp. 278 - 289, 2003.
5. T. Fahle, “Simple and Fast: Improving a Branch-and-Bound Algorithm for Maximum Clique”, Proceedings of the 10th European Symposium on Algorithms - ESA’02, Springer-LNCS 2461, pp. 485 - 498, 2002.
6. K. Satoh, “A Method for Generating Data Abstraction Based on Optimal Clique Search”, Master’s Thesis, Graduate School of Eng., Hokkaido Univ., March, 2003. (in Japanese)
7. S. Masuda, “Analysis of Ascidian Gene Expression Data by Clique Search”, Master’s Thesis, Graduate School of Eng., Hokkaido Univ., March, 2005. (in Japanese)
8. B. Shi, “Top-N Clique Search of Web Pages”, Master’s Thesis, Graduate School of Eng., Hokkaido Univ., March, 2005. (in Japanese)
9. Y. Okubo and M. Haraguchi, “Creating Abstract Concepts for Classification by Finding Top-N Maximal Weighted Cliques”, Proceedings of the 6th International Conference on Discovery Science - DS’03, Springer-LNAI 2843, pp. 418 - 425, 2003.
10. Y. Okubo and M. Haraguchi, “Finding Top-N Pseudo-Cliques in Simple Graph”, Proceedings of the 9th World Multiconference on Systemics, Cybernetics and Informatics - WMSCI’05, Vol. III, pp. 215 - 220, 2005.

CLASSIC'CL: An Integrated ILP System

Christian Stolle, Andreas Karwath, and Luc De Raedt

Albert-Ludwigs Universität Freiburg, Institut für Informatik,
Georges Köhler Allee 79, D-79110 Freiburg, Germany
{stolle, karwath, deraedt}@informatik.uni-freiburg.de

Abstract. A novel inductive logic programming system, called *Classic'cl* is presented. *Classic'cl* integrates several settings for learning, in particular learning from interpretations and learning from satisfiability. Within these settings, it addresses descriptive and probabilistic modeling tasks. As such, *Classic'cl* (C-armr, cLAudien, icl-S(S)at, ICl, and CLl-pad) integrates several well-known inductive logic programming systems such as Claudien, Warmr (and its extension C-armr), ICL, ICL-SAT, and LLPAD. We report on the implementation, the integration issues as well as on some experiments that compare *Classic'cl* with some of its predecessors.

1 Introduction

Over the last decade, a variety of ILP systems have been developed. At the same time, some of the most advanced systems such as Progol [12, 16] and ACE [3] can solve several different types of problems or problem settings. ACE induces rules (as in ICL [7]), decision trees (as in TILDE [1]) and frequent patterns and association rules (as in Warmr [8]). However, most of the present ILP techniques focus on predictive data mining setting and also deal with the traditional learning from entailment setting [4]. The key contribution of this paper is the introduction of the system *Classic'cl*, which learns from interpretations in a descriptive setting. The key novelty is that it tightly integrates several descriptive ILP, such as Claudien [5], Warmr [8], C-armr [6], and LLPADs [15]. This is realized using a generalized descriptive ILP algorithm that employs conjunctive constraints for specifying the clauses of interest. A wide variety of constraints is incorporated, including minimum frequency, exclusive disjunctions, and condensed representations [6]. By combining constraints in different ways,*Classic'cl* can emulate Warmr, Claudien, C-armr and LLPADS as well as some novel variations. *Classic'cl* is derived from the implementation of C-armr [6]. The performance of *Classic'cl* is experimentally compared with some of its predecessors, such as ACE and Claudien. In addition to the descriptive setting, *Classic'cl* also includes a predictive learning setting that emulates the ICL system [7]. This setting is not covered in this paper.

This paper relies on some (inductive) logic programming concepts. The reader unfamiliar with this terminology is referred to [13] for more details.

In the following section we introduce general constraints for the descriptive ILP problem and show how known algorithms can be expressed in this formalism.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 354–362, 2005.

A general algorithm to tackle this problem is presented in 3, some implementational issues are described in section 4 and experiments are presented in section 5. We conclude in section 6.

2 The Descriptive ILP Problem

2.1 Constraint Based Mining Problem

Mannila and Toivonen [11] formalized the task of data mining as that of finding the set $Th(Q, D, \mathcal{L})$, where Q is a constraint or query, D a data set and $\mathcal{L}$ a set of patterns. $Th(Q, D, \mathcal{L})$ then contains all patterns h in $\mathcal{L}$ that satisfy the constraint Q w.r.t. the data set D, i.e. $Th(Q, D, \mathcal{L}) = \{h \in \mathcal{L} | Q(h, D) = true\}$. When applying this definition of descriptive data mining to ILP, the language $\mathcal{L}$ will be a set of clauses, the data set D a set of examples and Q can be a complex constraint. Clauses are expressions of the form $h_1 \vee \cdots \vee h_n \leftarrow b_1 \wedge \cdots \wedge b_m$ where the h_i and b_j are logical atoms and all variables are universally quantified (cf. appendix in [13]). The learning from interpretations setting is incorporated by many well-known systems such as Claudien, Warmr, C-armr, Farmr, and LLPADS. We therefore choose interpretations as examples. In this paper, an interpretation is a set of ground facts. The above leads to the descriptive ILP problem, which is tackled in this paper:

Given:
- a language $\mathcal{L}_h$ (i.e., a set of clauses)
- a set of interpretations E
- a constraint $cons(h, E) \in \{true, false\}$ where $h \in \mathcal{L}_h$

Find:
- $Th(cons, E, \mathcal{L}_h)$, i.e., the set of clauses $c \in \mathcal{L}_h$ for which $cons(c, E) = true$

Using this generic formulation of descriptive ILP, we can now consider various constraints *cons* as a conjunction of constraints $c_1 \wedge \cdots \wedge c_k$ (e.g frequency, covers, cf. below). Some of the constraints can be monotonic or anti-monotonic, which can be used to prune the search space. A constraint $cons_m$ is monotonic if all specializations of a clause c will satisfy $cons_m$ whenever c does, and a constraint $cons_a$ is anti-monotonic if all generalizations of a clause c will satisfy $cons_m$ whenever c does. As framework for generality we employ Plotkin's θ-subsumption, which is the standard in ILP. It states that a clause c is more general than a clause c' if and only if there exists a substitution θ such that $c\theta \subset C'$.

2.2 Constraints for ILP

Motivated by constraints used in Claudien, Warmr, C-armr, and LLPAD, *Classic'cl* employs constraints defined on clauses of the form $h_1 \vee \cdots \vee h_n \leftarrow b_1 \wedge \cdots \wedge b_m$:

1. *query* is *true* iff the head of the clause is empty, i.e., if $n = 0$. This constraint is built-in in systems searching for frequent queries such as Warmr and C-armr.

2. *covers*(e) is *true* for an interpretation $e \in E$ iff $\leftarrow b_1 \wedge \cdots \wedge b_m$ succeeds in e, i.e. if there is a substitution θ s.t. $\{b_1\theta, \ldots, b_m\theta\} \subseteq e$. E.g., $\leftarrow drinks(X), beer(X)$ covers $\{drinks(vodka)$, $liquor(vodka)$, $drinks(duvel)$, $beer(duvel)\}$. This constraint is often used in the case of queries (i.e., where $n = 0$).
3. *satisfies*(e) is *true* iff $h_1 \vee \cdots \vee h_n \leftarrow b_1 \wedge \cdots \wedge b_m$ satisfies $e \in E$, i.e., iff $\forall\theta$: $\{b_1\theta, \ldots, b_m\theta\} \subseteq e \rightarrow \{h_1\theta, \ldots, h_n\theta\} \cap e \neq \emptyset$, e.g. the clause $beer(X) \leftarrow drinks(X)$ does not satisfy the interpretation $\{drinks(vodka)$, $liquor(vodka)$, $drinks(duvel)$, $beer(duvel)\}$ but does satisfy $\{drinks(duvel)$, $beer(duvel)\}$.
4. *xor*(e) is *true* iff for any two $h_i \neq h_j$ there exist no substitutions θ_1 and θ_2 such that $\{b_1\theta_1, \ldots, b_m\theta_1, h_i\theta_1\} \subseteq e$ and $\{b_1\theta_2, \ldots, b_m\theta_2, h_j\theta_2\} \subseteq e$. The *xor* constraint specifies that at most one literal in the head of the clause can be *true* within the interpretation e.
5. $freq(cons, E) = |\{e \in E | cons(e)\}|$ specifies the number of examples e in E for which the constraint $cons(e)$ is *true*. This is typically used in combination with the constraints *satisfies* or *covers*.
6. *maxgen* is *true* iff $h_1 \vee \cdots \vee h_n \leftarrow b_1 \wedge \cdots \wedge b_m$ satisfies the monotonic part of the rest of the constraint *cons* and no clause $h_1 \vee \cdots \vee h_{i-1} \vee h_{i+1} \vee \cdots \vee h_n \leftarrow b_1 \wedge \cdots \wedge b_m$ satisfies *cons*. This constraint is needed as there may be an infinite number of refinements of such clauses that satisfy a monotonic constraint.
7. *s-free(T)* is *true*, where T is a set of horn clauses, iff there is no range-restricted clause $p \leftarrow b'_1 \wedge \cdots \wedge b'_k$ where all $b'_i \in \{b_1, \ldots, b_m\}$ and $p \in \{b_1, \ldots, b_m\} - \{b'_1 \wedge \cdots \wedge b'_k\}$ for which $T \models p \leftarrow b'_1 \wedge \cdots \wedge b'_k$. So no redundancies are induced w.r.t. a background theory T that specifies properties of the predicates (cf. [6]). E.g. $T = \{leq(X, Z) \leftarrow leq(X, Y), leq(Y, Z)\}$ (transitivity) averts clauses such as ($\leftarrow leq(X, Y)$, $leq(Y, Z), leq(X, Z)$) as the last literal is redundant.
8. *free*(E) is *true* iff there is no range-restricted clause $p \leftarrow b'_1 \wedge \cdots \wedge b'_k$ where all $b'_i \in \{b_1, \ldots, b_m\}$ and $p \in \{b_1, \ldots, b_m\}$ and $p \neq b_i$ for which $freq(p \leftarrow b'_1 \wedge \cdots \wedge b'_k, satisfies, E) = |E|$. This assures that there are no redundant literals given the data. E.g., given the interpretation $I := \{beer(duvel)$, $alcohol(duvel)$, $alcohol(vodka)\}$, the clause $\leftarrow beer(X)$ is free while $\leftarrow beer(X) \wedge alcohol(X)$ is not free, as the clause $alcohol(X) \leftarrow beer(X)$ is satisfied by I (cf. [6]).
9. *δ-free(E)* is *true*, where δ is a natural number, iff there is no range-restricted clause $p \leftarrow b'_1 \wedge \cdots \wedge b'_k$ where all $b'_i \in \{b_1, \ldots, b_m\}$ and $p \in \{b_1, \ldots, b_m\} - \{b'_1 \wedge \cdots \wedge b'_k\}$ for which $freq(p \leftarrow b'_1 \wedge \cdots \wedge b'_k, satisfies, E) \geq |E| - \delta$. It is not required that the rule perfectly holds on the data, but only that it holds approximately, as δ exceptions are allowed (cf. [6]).
10. *consistent*(T) is *true*, where T is a set of horn clauses, if and only if $T \cup \{h_1 \vee \cdots \vee h_n \leftarrow b_1 \wedge \cdots \wedge b_m\} \not\models \Box$, i.e., if it is satisfiable. E.g., consider the theory $T = \{\leftarrow parent(X, X)\}$ which specifies that no one is its own parent. Any clause containing this literal is not consistent with respect to T.

The above specified constraints have the following properties: $freq(h, cons_m, E) > t$ and *satisfies* are monotonic, while *covers*, *query*,

$consistent$, $s-free$, $free$, δ-free, and $freq(h, cons_a, E) > t$ are anti-monotonic. xor is anti-monotonic w.r.t. the head only, i.e., xor is anti-monotonic w.r.t. a fixed body. Clauses with an empty head always satisfy the xor constraint. Therefore, this constraint only applies when refining the heads of clauses. The $maxgen$ constraint is is neither monotonic nor anti-monotonic. Therefore, it will require special attention in our algorithm.

2.3 Existing Descriptive ILP Systems

Claudien [5] essentially searches for all maximally general clauses that satisfy a set of interpretations. This corresponds to using the constraint $cons = maxgen \wedge freq(satifies, E) = |E|$. E.g., given the interpretation $I = \{vodka(smirnov)$, $beer(duvel)$,$alcohol(smirnov)$, $alcohol(duvel)\}$ and a language bias over the literals in I, one would find the following clauses: $\{beer(X) \vee vodka(X) \leftarrow alcohol(X)$; $\leftarrow beer(X) \wedge vodka(X)$; $alcohol(X) \leftarrow vodka(X)$; $alcohol(X) \leftarrow beer(X)\}$.

Warmr [8] extends the well-known Apriori system to a relational data mining setting. It employs essentially the constraints $cons = query \wedge freq(covers, E) > t$. In the example above ($t = 1$) these queries would be generated: $\{\leftarrow beer(X)$; $\leftarrow vodka(X)$; $\leftarrow alcohol(X)$; $\leftarrow beer(X) \wedge alcohol(X)$; $\leftarrow vodka(X) \wedge alcohol(X)\}$.

C-armr [6] is a variant of Warmr that extends Warmr with condensed representations. Additional constraints that can be imposed include $free$, $s - free$, $consistent$ and $\delta - free$. On the same example, and having the additional constraint $free$, the following queries would be generated. $\{\leftarrow beer(X); \leftarrow vodka(X); \leftarrow alcohol(X)\}$.

CLLPAD combines ideas from Claudien with probabilistic ILP. It essentially mines for LPADS, [17]. These consists of annotated clauses of the form $(h_1 : \alpha_1) \vee \cdots \vee (h_n : \alpha_n) \leftarrow b_1 \wedge \cdots \wedge b_m$. The $\alpha_i \in [0, 1]$ are real-valued numbers, s.t. $\sum_{i=1}^{n} \alpha_i = 1$. The head atoms h_i of the clauses fulfill the xor constraint, such that for each interpretation at most one h_i is $true$ with a certain probability. This ensures that the clauses c_i of an LPAD P can be considered independently as in traditional inductive logical programs.

$$cons = maxgen \wedge \bigwedge_{e \in E} xor(e) \wedge freq(satisfies, E) = |E| \wedge freq(covers, E) \geq 1$$

Notice that the xor constraint together with $satisfies$ actually implies $maxgen$, so that the CLLPAD can be considered a specialization of the Claudien setting. This constraint is imposed in an early system inducing LPADs, LLPAD [15]. The annotated clauses satisfying $cons$ are then composed to LPADs in a post-processing step (cf. [15]). E.g., consider the following interpretations $\{beer(duvel)$, $alcohol(duvel)\}$ and $\{vodka(smirnov)$, $alcohol(smirnov)\}$. The clauses $\{0.5 : vodka(X) \vee$ $0.5 : beer(X) \leftarrow alcohol(X)$; $1.0 : alcohol(X) \leftarrow vodka(X)$; $1.0 : alcohol(X) \leftarrow beer(X)\}$ would satisfy the constraints. As in [15] the rules get annotated using the equation $\alpha_i = \frac{\sum_{e \in E, satisfies(h_i \leftarrow b_1 \wedge \cdots \wedge b_n, e)} \pi_P^*(e)}{\sum_{e \in E, covers(\leftarrow b_1 \wedge \cdots \wedge b_n, e)} \pi_P^*(e)}$, where the $\pi_P^*(E)$ denotes the probabilities of the interpretations specified in the data set. So the probability of h_i is the sum of probabilities of the interpretations which are covered by $h_i \wedge b$ divided by the sum of probabilities of the interpretations which are covered by b.

The usage of these constraints opens the possibility for several new combinations:

- introduction of condensed representations within the Claudien and CLLPAD setting. The effect of constraints as $free$, $\delta - free$, and $s - free$ is that less patterns are found, that they are typically found more efficiently, and also that (for $free$ and $s - free$) only redundant and undesirable clauses are pruned away, without affecting the semantics of the solution set.
- the original implementation of LLPAD, as described in [15], does not seem to allow for the use of variables in clauses, which essentially corresponds to a propositional version of LLPAD. In contrast, the version in*Classic'cl* does allow for variabelized clauses.
- new combinations, combining, e.g., $freq(satisfies, E)$, $freq(covers, E)$ and δ-free, now become possible.

3 The Descriptive ILP Algorithm

By now we are able to specify the algorithm. We will first discover all bodies that satisfy the constraints, and then expand these into those clauses that satisfy also the head. The algorithm employs two different phases for realizing that. The first phase employs a body refinement operator ρ_b, which merely refines the body of a clause whereas the second phase employs a head refinement operator ρ_o, which merely refines the head by adding literals to the conclusion part of clauses.

Algorithm 1 The generic function **body**$(cons, E)$.

```
$C_0 := \{false \leftarrow true\}$; $i := 0$; $F_0 := I_0 := \emptyset$
while $C_i \neq \emptyset$ do
  $F_i := \{c \in C_i | cons_a(c, E)\}$
  if $cons$ does not contain the constraint $query$ then
    call head$(cons, F_i)$
  else
    output $\{f \in F_i | cons_m(f, E)\}$
  end if
  $I_i := C_i - F_i$
  $C_{i+1} := \{b' \mid b \in F_i$ and $b' \in \rho_b(b)$ and $\neg\exists s \in \bigcup_j I_j : s \preceq b'\}$
  $i := i + 1$
end while
```

The *body* function (algorithm 1) is very similar to a traditional level wise search algorithm such as Warmr. It starts from the empty query and repeatedly refines it – in a level wise fashion – until the anti-monotonic $cons_a$ part of the constraint *cons* no longer holds on candidate clauses. The algorithm does not only keep track of the clauses satisfying the anti-monotonic constraint $cons_a$ (on the F_i) but also of the negative border (using the I_i). This is useful for pruning because – when working with a language bias specified using rmodes (cf. below) – not all clauses in the θ-subsumption lattice are within the language $\mathcal{L}_h$, i.e. the language $\mathcal{L}_h$ is not anti-monotonic. Consider for instance the clause $p(K) \leftarrow benzene(K, S) \wedge member(A, S) \wedge atom(K, A, c)$. Even though this clause will typically satisfy the syntactic constraints, its generalization $p(K) \leftarrow member(A, S)$ will typically not be mode-conform. Furthermore, when a new candidate is generated, it is tested whether the candidate is not subsumed by an already known infrequent one.

Algorithm 2 The generic function **head**$(cons, F)$.

```
$C_0 := F$; $i := 0$; $S_0 := I_0 := \emptyset$
while $C_i \neq \emptyset$ do
  $S_i := \{c \in C_i | cons_m(c, E)\}$
  if $cons$ does contain the constraint $maxgen$ then
    $I_i := C_i - S_i$
    $S_i := \{c \in S_i | \neg\exists s \in \bigcup_j S_j : s \preceq c\}$
  else
    $I_i := C_i$
  end if
  $C_{i+1} := \{c' \ | c \in I_i$ and $c' \in \rho_h(c)$ and $cons_a(c', E), \}$
  $i := i + 1$
end while
output $filter(\cup_i S_i)$
```

The interesting and new part of the algorithm is concerned with the function *head* (algorithm 2). This part is used if $query \notin cons$, and one searches for proper clauses, not just queries. The algorithm then proceeds as follows. The *head* function is invoked using the call $head(cons, F)$ for every body. Within the procedure only the head is changed using a head refinement operator ρ_h (which adds literals to the head). Within this context, the algorithm *head* is similar in spirit to the level wise algorithm, except that if the constraint *maxgen* is included in *cons*, those clauses that satisfy *cons* are no longer refined. The algorithm employs a list of candidate clauses on C_i. Those candidates satisfying the constraint are put on S_i, the set of solutions. Depending on *maxgen* all candidates on C_i or only those not satisfying *cons* are refined. The algorithm then outputs, according to some output filter (e.g. a filter that annotates the clauses for CLLPAD), all solutions $\cup S_i$.

4 Implementation Issues

Language Bias. Within ILP, $\mathcal{L}_h$ typically imposes syntactic restrictions on the clauses to be used as patterns. Whereas some of the original implementations (such as Claudien [5]) employed complex formalisms such as DLAB, *Classic'cl* uses the now standard mode and type restrictions (rmodes) of ILP.

Optimizations and Optimal Refinement Operators. In order to search efficiently for solutions, it is important that each relevant pattern is generated at most once. For this, optimal refinement operators (using some canonical form) are employed. As *Classic'cl* is based on the original C-armr implementation of [6], it employs the same optimal refinement operator. In a similar way, we have used a canonical form and optimal refinement operator defined for disjunctive head literals with a fixed body. As computing constraints like frequency are computationally expensive, we have employed the same optimizations as in [6], the system is equally designed as a *light* Prolog implementation that is small but still reasonably efficient.

5 Experiments

The aim was to 1) investigate the performance of *Classic'cl* w.r.t the original implementations, and 2) show that we can tackle some new problem settings.

Datasets. We used artificial and real-world datasets. As artificial datasets, we used the Bongard 300 and 6013 datasets. As real world datasets, we have chosen the Mutagenesis data set [10], the secondary structure prediction dataset from [14], and the SCOP-fold dataset [9].

Warmr and C-armr. First, we compared ACE-Warmr with *Classic'cl.* ACE-Warmr is the original Warmr algorithm in the ACE toolkit [3]. ACE is implemented in a custom build Prolog (iProlog), and can be used with a number of optimizations, like query packs [2]. The results of the comparison can be seen in table 1. The different number of frequent patterns is due to a slightly different language bias and operators. If one takes as criterion time per pattern, then ACE-Warmr and *Classic'cl* are more or less comparable in this experiment.

As a second test, we investigated searching for disjunctive clauses versus searching for horn clauses. This compares to the settings $cons_1 = freq(h, covers, E) > t \wedge query(h) \wedge freq(h, satisfies, E) > t$ to $cons_2 = query(h) \wedge freq(h, covers, E) > t$.

Claudien. We evaluated Classic'cl Claudien compared to the original Claudien implementation using the Mutagenisis and Bongard datasets. All tests we ran on a SUN Blade 1550, as we only had a compiled version for the original Claudien version available. We only mined for horn clauses with a maximum of 5 literals in the Mutagenesis case. This was necessary, as the computational costs proved to be too expensive for the original Claudien. In the case of the Bongard 300 experiment we also restricted the search to definite clauses, as the language bias definition languages rmodes and DLAB are too different to generate comparable results. The results can be found in table 3.

CLLPAD. We employed the LPAD setting and applied it to the SCOP dataset. The test was to evaluate the applicability of the CLLPAD setting to a real world

Table 1. Comparison between the ACE WARMR and *Classic'cl* in the Warmr and C-armr setting on mutagenesis. For the C-armr setting, we chose to employ $\delta - free$, $s - free$, *consistent* (with $\delta = 0$, t = 2 and maxlevel = 4). ACE-Warmr (packs) denotes the setting for ACE with the option 'use_packs(ilp)'.

	Runtime [secs].	# freq. Patterns
ACE-Warmr(no packs)	12960	91053
ACE-Warmr(packs)	1816	91053
***Classic'cl*-Warmr**	5301	194737
***Classic'cl*-Carmr()**	4622	124169

Table 2. Comparison between the run times and number of rules for the definite ($cons = query(h) \wedge freq(h, covers, E) > t$) and disjunctive ($cons = query(h) \wedge freq(h, satisfies, E) > t$) search

		Runtime [s]		**# Rules**		**Factor**	
Data set	**Subset**	**Horn**	**Disj.**	**Horn**	**Disj.**	**Horn**	**Disj**
Mutagenesis	188	2602.62	4098.26	893	9099	1.57	10.19
	42	1454.52	1839.45	996	6291	1.26	6.32
	230	3484.94	5339.67	1002	9904	1.53	9.88
Bongard	300	4.78	12.52	54	1628	2.62	30.15
	6013	212.02	1597.97	114	2610	7.54	22.89
Sec. Structure	alpha	75414.4	76950.51	1188	18145	1.02	15.27
	beta	162.79	188.11	111	16768	1.16	151.06
	coil	55102.04	55827.35	1186	18146	1.01	15.3

Table 3. Comparison between the original Claudien and the Classic'cl in the Claudien setting. The differences in the number of rules found is due to the different language bias used (DLAB vs. rmodes). To avoid the comparison between the different setting we also present the time spent by the two implementations producing a rule in seconds per rule. Classic'cl clearly outperforms the original algorithm.

			Runtime [s]		**# Rules**		**Sec. p. rule**		**Factor**
Dataset	**Subset**	**Level**	**Orig.**	**Classic**	**Orig.**	**Classic**	**Orig.**	**Classic**	
Mutagenesis	188	4	66631.9	3290.6	262	308	254.32	10.68	23.8
	42	4	12964.3	1214.41	123	303	105.40	4.01	26.3
	230	4	86022.3	4490.62	279	418	308.32	10.74	28.7
Bongard	300	5	71.53	14.44	32	51	2.24	0.28	7.89

database. The initial set of clauses, *Classi'cl* took 5,714 seconds to construct. Applying the post processing filter solving the CSP took 5,742 seconds and resulted in 33 LPADs build from 18 horn clauses and 7 annotated disjunctive clauses. The disjunctive clauses produced, all center around three folds, name fold1, fold37, and fold55. For space limitations, detailed results are omitted from this paper. This application was impossible with the previous implementation of LLPADs which only employes propositional examples.

To summarize, the experiments clearly show that Classic'cl can indeed simulate its predecessors, that its performance is much better of that of Claudien and despite the light Prolog implementation realistic enough to be applied to real-world data.

6 Conclusions

A novel descriptive data mining approach within the ILP setting of learning from interpretations has been presented. The approach incorporates ideas from constraint based mining in that a rich variety of constraints on target hypotheses can be specified. The algorithm is also incorporated in the system*Classic'cl*, which is able to emulate many of its predecessors such as Claudien, Warmr, c-Armr, CLLPad, as well as ICL and ICL-SAT, as well as some new settings. Classic'cl is implemented in Prolog and it is available from the authors.

References

[1] H. Blockeel and L. De Raedt. Top-down induction of first order logical decision trees. *AI*, 101(1-2):285–297, 1998.

[2] H. Blockeel, B. Dehaspe, L.and Demoen, and G. Janssens. Improving the efficiency of inductive logic programming through the use of query packs. *JAIR*, 16:135–166, 2002.

[3] H. Blockeel, L. Dehaspe, J. Ramon, and J.Struyf. Ace - a combined system.

[4] L. De Raedt. Logical settings for concept-learning. *AI*, 95(1):187–201, 1997.

[5] L. De Raedt and L. Dehaspe. Clausal discovery. *Machine Learning*, 26:99–146, 1997.

[6] L. De Raedt and J. Ramon. Condensed representations for inductive logic programming. In *KR '04*, pages 438–446, 2004.

[7] L. De Raedt and W. Van Laer. Inductive constraint logic. In *ALT95*, volume 997 of *LNAI*. SV, 1995.

[8] L. Dehaspe. *Frequent Pattern Discovery in First-Order Logic.* K. U. Leuven, Dec. 1998.

[9] K. K. Kersting, T. Raiko, S. Kramer, and L. De Raedt. Towards discovering structural signatures of protein folds based on logical hidden markov models. In *PSB 2003*, pages 192–203, 2003.

[10] R. D. King, S. Muggleton, A. Srinivasan, and M. J. E. Sternberg. Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *PNAS*, 93:438–442, 1996.

[11] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. In *Data Mining and Knowledge Discovery*, volume 1, 1997.

[12] S. Muggleton. Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4):245–286, 1995.

[13] S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic programming*, 17(20):629–679, 1994.

[14] S. Muggleton, R. D. King, and M. J. E. Sternberg. Protein secondary structure prediction using logic. In S. Muggleton, editor, *ILP'92*, Report ICOT TM-1182, pages 228–259, 1992.

[15] F. Riguzzi. Learning logic programs with annotated disjunctions. In *ILP'04*, 2004.

[16] A. Srinivasan. The aleph manual.

[17] J. Vennekens, S. Verbaeten, and M. Bruynooghe. Logic programs with annotated disjunctions. Technical report cw386, K.U. Leuven, 2003.

Detecting and Revising Misclassifications Using ILP

Masaki Yokoyama, Tohgoroh Matsui, and Hayato Ohwada

Department of Industrial Administration, Faculty of Science and Technology,
Tokyo University of Science,
2641 Yamazaki, Noda-shi, Chiba 278-8510, Japan
j7404659@ed.noda.tus.ac.jp
{matsui, ohwada}@ia.noda.tus.ac.jp

Abstract. This paper proposes a method for detecting misclassifications of a classification rule and then revising them. Given a rule and a set of examples, the method divides misclassifications by the rule into *miscovered* examples and *uncovered* examples, and then, separately, learns to detect them using Inductive Logic Programming (ILP). The method then combines the acquired rules with the initial rule and revises the labels of misclassified examples. The paper shows the effectiveness of the proposed method by theoretical analysis. In addition, it presents experimental results, using the Brill tagger for Part-Of-Speech (POS) tagging.

1 Introduction

Classification is one of the most popular fields in machine learning. It is concerned with constructing new classification rules from given training examples. Most previous work has focused on creating rules from scratch. Therefore, these approaches do not make use of previously constructed classification rules, even if they are reasonable. We consider that such rules are useful, and that it is more effective to correct misclassifications of a rule, than to create a new classification rule from scratch.

In this paper, we propose a method that detects misclassifications of a classification rule and then revises them. Given a rule and a set of examples, the method divides misclassifications by the rule into *miscovered* examples and *uncovered* examples and, separately, learns to detect them. It then combines the acquired rules with the initial rule and revises the labels of misclassified examples. This paper shows the effectiveness of the proposed method by theoretical analysis.

We use Inductive Logic Programming (ILP) to learn rules for detecting and revising misclassifications. ILP is a framework that combines machine learning and logic programming. ILP systems construct logic programs from examples and from background knowledge, which is also described by logic programs. One of the most important advantages of using ILP for discovering knowledge is that ILP can acquire hypotheses that can be understood by human beings. Another important advantage of ILP is that it is able to use background knowledge.

We have applied our method to Part-Of-Speech (POS) tagging, to which ILP has been applied previously [1]. We use the Brill tagger [2] as the initial classifier, which is one of the best rule-based tagging systems and is widely used in research into natural

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 363–370, 2005.

language processing. This paper shows the results of combining the Brill tagger with the additional acquired rules.

2 Miscovered Examples and Uncovered Examples

In this paper, we consider binary classification, which is also called concept learning. Let x be an example from a set of possible examples $\mathcal{X}$. The example is expressed as $(x, c(x))$, where c is a target function. If x belongs to the target concept, then $c(x) = 1$; if otherwise, $c(x) = 0$.

Misclassified examples of a classification rule are either *miscovered* examples or *uncovered* examples. Consider a classification rule r. Let h_{r} be the hypothesis function of r: if it estimates that x belongs to the target concept, then $h_{\mathrm{r}}(x) = 1$; otherwise, $h_{\mathrm{r}}(x) = 0$. We say that an example $x \in \mathcal{X}$ is *miscovered* by a classification rule r whenever $c(x) = 0$, but $h_{\mathrm{r}}(x) = 1$. We say that x is *uncovered* by r whenever $c(x) = 1$, but $h_{\mathrm{r}}(x) = 0$. Fig. 1 shows miscovered examples and uncovered examples of a classification rule r for a target concept. Miscovered examples and uncovered examples are sometimes called false positives and false negatives, respectively.

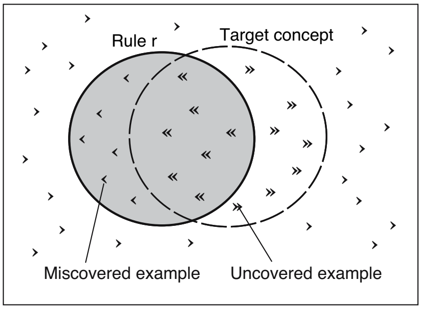


Fig. 1. Miscovered examples and Uncovered examples of a Classification Rule r for a Target concept

3 Method

3.1 Detecting and Revising Miscovered Examples

First, we consider the detection and revision of miscovered examples by using ILP. We generate examples for ILP from the data set by using the initial classification rule. We then construct a rule for detecting miscovered examples. Finally, we revise the labels of the detected miscovered examples.

Consider a classification rule r. Because all of the examples miscovered by r are included in examples covered by r, we can define the problem of detecting miscovered examples as follows: given a classification rule r and an example x that is covered by r, estimate whether x is miscovered or not.

Denote the subset of training examples that are covered by r as $\mathcal{E}_{\mathrm{m}}$. We then divide them into miscovered and correctly covered examples. Let $\mathcal{E}_{\mathrm{m}}^{+}$ be the set of miscovered examples, and let $\mathcal{E}_{\mathrm{m}}^{-}$ be the set of correctly covered examples. $\mathcal{E}_{\mathrm{m}}^{+}$ and $\mathcal{E}_{\mathrm{m}}^{-}$ can be written as:

$$\mathcal{E}_{\mathrm{m}}^{+} = \{x \mid (x, c(x)) \in \mathcal{D},\ h_{\mathrm{r}}(x) = 1,\ c(x) = 0\},$$
$$\mathcal{E}_{\mathrm{m}}^{-} = \{x \mid (x, c(x)) \in \mathcal{D},\ h_{\mathrm{r}}(x) = 1,\ c(x) = 1\},$$

where $\mathcal{D}$ is the set of training examples, h_{r} is the estimating function of r, and c is the target-concept function. This is shown in the left hand figure in Fig. 2, where the $+$ signs are positive examples and $-$ signs are negative examples.

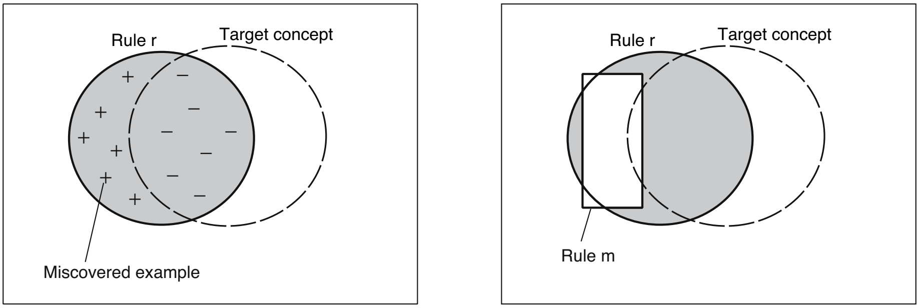


Fig. 2. Training examples for the miscovered concept (left) and the combined classification rule, h_{rm}, of the acquired rule and the initial rule (right)

Next, using ILP, we acquire a hypothesis $\mathcal{H}_{\mathrm{m}}$ from $\mathcal{E}_{m}^{+}$, $\mathcal{E}_{m}^{-}$, and background knowledge $\mathcal{B}$, such that $\mathcal{B} \vee \mathcal{H}_{\mathrm{m}} \models \mathcal{E}_{m}^{+}$ and $\mathcal{B} \vee \mathcal{H}_{\mathrm{m}} \not\models \mathcal{E}_{m}^{-}$. We define the estimating function h_{m} as: if $\mathcal{B} \vee \mathcal{H}_{\mathrm{m}} \models x$ for an example $x \in X$, then $h_{\mathrm{m}}(x) = 1$; otherwise, $h_{\mathrm{m}}(x) = 0$.

After acquiring $\mathcal{H}_{\mathrm{m}}$, we revise the misclassified labels by combining h_{r} with h_{m}. We define the combined hypothesis function h_{rm} as:

$$h_{\mathrm{rm}}(x) = \begin{cases} 1 & \text{if } h_{\mathrm{r}}(x) = 1 \text{ and } h_{\mathrm{m}}(x) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

The right-hand figure of Fig. 2 illustrates this combined classification rule rm. If an example is included in the shaded area, the classification rule now estimates that it belongs to the target concept.

3.2 Detecting and Revising Uncovered Examples

We now consider uncovered examples. Again, we generate examples for detection and then revision. Previously, we used examples covered by r as a source of miscovered examples, but now we use the remaining examples, i.e., examples not covered by r. Denote the subset of training examples that are not covered by r as $\mathcal{E}_{\mathrm{u}}$. We divide these

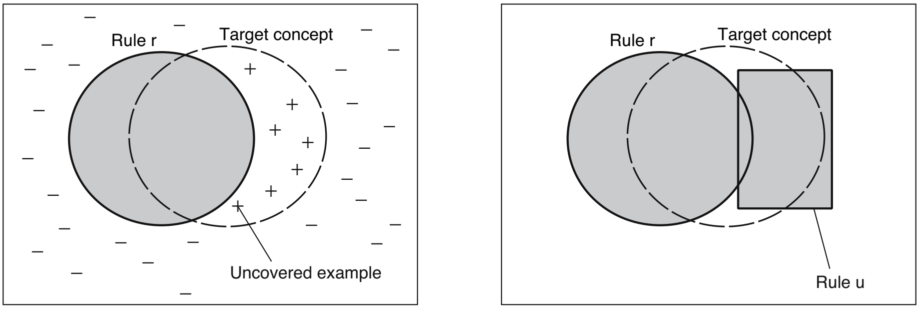


Fig. 3. Training examples for the uncovered concept (left) and the combined classification rule, h_{ru}, of the acquired rule and the initial rule (right)

examples into two subsets. Let $\mathcal{E}_{\mathrm{u}}^{+}$ be the set of uncovered examples, and let $\mathcal{E}_{\mathrm{u}}^{-}$ be the set of correctly not-covered examples. $\mathcal{E}_{\mathrm{u}}^{+}$ and $\mathcal{E}_{\mathrm{u}}^{-}$ can be written as:

$$\mathcal{E}_{\mathrm{u}}^{+} = \{x \mid (x, c(x)) \in \mathcal{D},\ h_{\mathrm{r}}(x) = 0,\ c(x) = 1\}\,,$$
$$\mathcal{E}_{\mathrm{u}}^{-} = \{x \mid (x, c(x)) \in \mathcal{D},\ h_{\mathrm{r}}(x) = 0,\ c(x) = 0\}\,.$$

The left-hand figure of Fig. 3 shows these training examples $\mathcal{E}_{\mathrm{u}}^{+}$ and $\mathcal{E}_{\mathrm{u}}^{-}$.

We now construct a hypothesis $\mathcal{H}_{\mathrm{u}}$ from $\mathcal{E}_{\mathrm{u}}^{+}$, $\mathcal{E}_{\mathrm{u}}^{-}$, and background knowledge $\mathcal{B}$, using ILP. We define the estimating function as h_{u}: $h_{\mathrm{u}}(x) = 1$ if $\mathcal{B} \vee \mathcal{H}_{\mathrm{u}} \models x$ for an example $x \in X$; otherwise, $h_{\mathrm{u}}(x) = 0$. After acquiring $\mathcal{H}_{\mathrm{u}}$, we revise the misclassified labels by combining h_{r} with h_{u}. We define the combined hypothesis function h_{ru} as:

$$h_{\mathrm{ru}}(x) = \begin{cases} 1 & \text{if } h_{\mathrm{r}}(x) = 1 \text{ or } h_{\mathrm{u}}(x) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The right-hand figure of Fig. 3 illustrates this classification rule ru.

3.3 Detecting and Revising Misclassified Examples

Finally, we combine the two acquired hypotheses with the initial classification rule. Because h_{m} and h_{u} are constructed from nonoverlapping training sets, we can combine them directly. We define a combined estimating function h_{rmu}:

$$h_{\mathrm{rmu}}(x) = \begin{cases} 1 & \text{if } h_{\mathrm{r}}(x) = 1 \text{ and } h_{\mathrm{m}}(x) = 0, \text{ or } h_{\mathrm{r}}(x) = 0 \text{ and } h_{\mathrm{u}}(x) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Fig. 4 illustrates this final combined classification rule h_{rmu}. Given an example x, we firstly compute $h_{\mathrm{r}}(x)$. If we find that $h_{\mathrm{r}}(x) = 1$, then we calculate $h_{\mathrm{m}}(x)$; otherwise, we calculate $h_{\mathrm{u}}(x)$. Thus, we choose the second classification rule depending on the situation, and it revises labels that were misclassified by the initial classification rule.

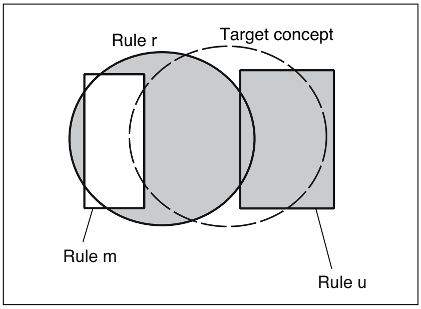


Fig. 4. The final combined classification rule h_{rmu}

4 Theoretical Analysis

We can show the effectiveness of the proposed method by theoretical analysis.

Theorem 1. *Let P_r and A_r be the precision and the accuracy of rule r. If the inequality $P_{\mathrm{m}} \geq 1/2$ is satisfied, then the inequality $A_{\mathrm{rm}} \geq A_{\mathrm{r}}$ is valid.*

Proof. To prove the theorem, consider the difference:

$$A_{\mathrm{rm}} - A_{\mathrm{r}} = \frac{|\mathrm{TP}_{\mathrm{rm}}| + |\mathrm{TN}_{\mathrm{rm}}|}{|\mathcal{E}_{\mathrm{rm}}|} - \frac{|\mathrm{TP}_{\mathrm{r}}| + |\mathrm{TN}_{\mathrm{r}}|}{|\mathcal{E}_{\mathrm{r}}|},$$

where $\mathcal{E}_{\mathrm{rm}}$ and $\mathcal{E}_{\mathrm{r}}$ are the example sets for rm and r, respectively. Since the example sets are the same, the denominators are the same, and positive. Now consider the numerators. In our method, examples classified by the rule rm can be written as:

$$\mathrm{TP}_{\mathrm{rm}} = \mathrm{TP}_{\mathrm{r}} \setminus \mathrm{FP}_{\mathrm{m}} \qquad \mathrm{FP}_{\mathrm{m}} \subseteq \mathrm{TP}_{\mathrm{r}}, \tag{1}$$

$$\mathrm{TN}_{\mathrm{rm}} = \mathrm{TN}_{\mathrm{r}} \cup \mathrm{TP}_{\mathrm{m}} \qquad \mathrm{TN}_{\mathrm{r}} \cap \mathrm{TP}_{\mathrm{m}} = \emptyset, \tag{2}$$

where TP_r, FP_r, FN_r, and TN_r are sets of true positive, false positive, false negative, and true negative examples of r, respectively. From Equations (1) and (2), the inequality

$$\begin{aligned}
&|\mathrm{TP}_{\mathrm{rm}}| + |\mathrm{TN}_{\mathrm{rm}}| - (|\mathrm{TP}_{\mathrm{r}}| + |\mathrm{TN}_{\mathrm{r}}|) \\
=& |\mathrm{TP}_{\mathrm{r}} \setminus \mathrm{FP}_{\mathrm{m}}| + |\mathrm{TN}_{\mathrm{r}} \cup \mathrm{TP}_{\mathrm{m}}| - (|\mathrm{TP}_{\mathrm{r}}| + |\mathrm{TN}_{\mathrm{r}}|) \\
=& (|\mathrm{TP}_{\mathrm{r}}| - |\mathrm{FP}_{\mathrm{m}}|) + (|\mathrm{TN}_{\mathrm{r}}| + |\mathrm{TP}_{\mathrm{m}}|) - (|\mathrm{TP}_{\mathrm{r}}| + |\mathrm{TN}_{\mathrm{r}}|) \\
=& |\mathrm{TP}_{\mathrm{m}}| - |\mathrm{FP}_{\mathrm{m}}| = |\mathrm{TP}_{\mathrm{m}}| \, \frac{2\,(P_{\mathrm{m}} - 1/2)}{P_{\mathrm{m}}} \geq 0
\end{aligned}$$

is valid, if the condition of the theorem is satisfied. The theorem is proved.

Theorem 2. *If the inequality $P_{\mathrm{u}} \geq 1/2$ is satisfied, then the inequality $A_{\mathrm{rmu}} \geq A_{\mathrm{rm}}$ is valid.*

This proof is omitted, to save space.

Finally, the following theorem indicates the effectiveness of our method:

Theorem 3. *If the inequalities* $P_{\mathrm{m}} \geq 1/2$ *and* $P_{\mathrm{u}} \geq 1/2$ *are satisfied, then the inequality* $A_{\mathrm{rmu}} \geq A_{\mathrm{r}}$ *is valid.*

Proof. From Theorems 1 and 2, $A_{\mathrm{rm}} \geq A_{\mathrm{r}}$ and $A_{\mathrm{rmu}} \geq A_{\mathrm{rm}}$ are valid, if the conditions of the theorem are satisfied. Therefore, the inequality $A_{\mathrm{rmu}} \geq A_{\mathrm{rm}} \geq A_{\mathrm{r}}$ is valid, if the conditions of the theorem are satisfied. The theorem is proved.

Since it is not difficult to learn a classifier whose precision is greater than or equal to $1/2$ in binary classification problems, the classification accuracy of our method can be higher than that of the initial classification rule.

5 Experiment: Part-of-Speech Tagging

5.1 Accuracy Comparison

POS tagging is the problem of assigning POS tags to each word in a document. We have applied our method to POS tagging, using the Brill tagger [2] as the initial classification rule. The data set is the set of Wall Street Journal articles in the Penn Treebank Project [3].

POS tagging involves more than three classes, and we adopted the one-against-the-rest method for formulation in terms of binary classification. Since there are 45 kinds of tags, we created 45 binary classification problems. For each problem, we applied the Brill tagger and created examples for learning the concepts of miscovered examples and uncovered examples. We used an ILP system, GKS [4,5], to learn the concepts with an acceptable error ratio of 0.2. We prepared the background knowledge of referring to the preceding three words and the following three words. We evaluated the performance of the acquired rules with 10-fold cross validation. We compared the accuracy of the initial classification rule of the Brill tagger with that of the proposed method. In this experiment, we added true-positive examples of the Brill tagger to the negative training examples for the uncovered concept. This enables us to acquire a hypothesis that covers only the uncovered examples. We also proved that Theorem 2 is true in this case.

Table 1 shows the results for each tag and overall. A_{r} stand for the accuracy of the Brill tagger alone. A_{rmu} stand for that of the combined classification rule, using the proposed method. P_{m} and P_{u} are the precisions of m and u alone, respectively. The "-" symbol means that the ILP system could not acquire rules at all. For all of the tags, the accuracies of the proposed method, A_{rmu}, were better than or equal to those of the Brill tagger alone, A_{r}. Because P_{m} and P_{u} were greater than $1/2$, the conditions of Theorem 3 were satisfied.

5.2 Discovered Knowledge on Misclassifications

There is another good aspect of the proposed method, in addition to increased accuracy: we have human-readable acquired knowledge on misclassifications, because ILP can create a hypothesis represented by first-order logic.

Here is the acquired knowledge for the "preposition" tag. The Prolog-formatted rule for the miscovered examples was as follows:

Table 1. The experiment result

Tag	A_{r}	A_{rmu}	P_{m}	P_{u}	Tag	A_{r}	A_{rmu}	P_{m}	P_{u}
cc	**0.9998**	**0.9998**	0.8889	-	pp	**0.9998**	**0.9999**	1.0	-
cd	**0.9991**	**0.9995**	1.0	0.9297	ppz	**0.9999**	**1.0**	-	1.0
cln	**0.9999**	**0.9999**	-	-	rb	**0.9947**	**0.9963**	0.9005	0.9488
cma	**0.9999**	**0.9999**	-	-	rbr	**0.9989**	**0.9992**	0.8682	0.9296
dlr	**1.0**	**1.0**	-	-	rbs	**0.9995**	**0.9999**	1.0	0.9482
dt	**0.9920**	**0.9988**	0.7778	0.9360	rp	**0.9984**	**0.9984**	-	-
ex	**0.9999**	**0.9999**	-	0.8472	rpn	**0.9988**	**0.9988**	-	-
fw	**0.9998**	**0.9999**	1.0	0.8710	rqt	**0.9999**	**0.9999**	0.8824	-
in	**0.9907**	**0.9943**	0.9947	0.9716	stp	**0.9999**	**0.9999**	-	-
jj	**0.9892**	**0.9924**	0.7888	0.9005	sym	**0.9987**	**0.9999**	-	0.9565
jjr	**0.9991**	**0.9993**	0.8788	0.8310	to	**0.9999**	**0.9999**	-	-
jjs	**0.9995**	**0.9996**	1.0	0.7640	uh	**0.9999**	**0.9999**	0.8000	-
lpn	**1.0**	**1.0**	-	-	vb	**0.9950**	**0.9974**	0.6429	0.8627
lqt	**1.0**	**1.0**	-	-	vbd	**0.9938**	**0.9949**	0.9162	0.9043
ls	**0.9999**	**0.9999**	-	-	vbg	**0.9976**	**0.9982**	0.6712	0.8708
md	**0.9999**	**0.9999**	-	-	vbn	**0.9924**	**0.9953**	0.7073	0.8614
nn	**0.9872**	**0.9914**	0.8165	0.9088	vbp	**0.9953**	**0.9965**	0.9888	0.9203
nns	**0.9967**	**0.9982**	0.8354	0.9133	vbz	**0.9971**	**0.9976**	0.9212	0.8766
np	**0.9941**	**0.9961**	0.7720	0.9401	wdt	**0.9976**	**0.9980**	0.9405	0.9730
nps	**0.9976**	**0.9978**	0.7024	0.8773	wp	**0.9999**	**0.9999**	-	-
pdt	**0.9998**	**0.9998**	0.8947	-	wpz	**1.0**	**1.0**	-	-
pnd	**1.0**	**1.0**	-	-	wrb	**0.9999**	**0.9999**	-	-
pos	**0.9986**	**0.9999**	-	0.9642	**All**	**0.9978**	**0.9986**	0.8973	0.9151

```
miscovered(A) :- post1word(A,'.').
miscovered(A) :- post2tag(A,vb), word(A,'like').
```

This rule means that the given word A is a miscovered example, i.e., it is not a preposition if: the following word is "." (period sign); or the next-but-one word is tagged "vb" and the given word is "like." Therefore, we can discover the Brill tagger mistakes with respect to prepositions. For example, the Brill tagger sometimes classifies the final word of a sentence as a preposition.

Similarly, we can see the rule for the uncovered examples. The rule is as follows:

```
uncovered(A) :- word(A,'up').
uncovered(A) :- post3word(A,'different').
```

This means that the given word A is an uncovered example, i.e. it is also a preposition if: the given word is "up", or the third-next word is "different".

We consider these rules to be very useful for correcting the Brill tagger itself. They show where we should change the Brill tagger's rule. So, if we install this knowledge into the Brill tagger, its performance will improve.

6 Conclusion

This paper proposes a method for decreasing misclassification, by using ILP to detect and revise misclassifications. The proposed method acquires two additional classification rules and combines them with the initial classification rule. We then show, by theoretical analysis, that this method works well. Finally, we apply it to POS tagging and present the experimental results.

Abney et al. have applied boosting to tagging [6]. They used their algorithm, AdaBoost, which calls a weak learner repeatedly to update the weights of examples. If the hypothesis acquired by the weak learner incorrectly classifies an example, it increases the weight; otherwise, it decreases the weight. Given an example to be predicted, boosting produces the final label, using a simple vote of the weak hypotheses. Although it can improve the classification accuracy very well, it cannot provide an understandable final hypothesis.

The good points of our method are that:

- it is simple and reliable,
- it can reduce the misclassification produced by the initial classification rule,
- it is shown that the classification accuracy of our method can be higher than that of initial classification rule, and
- the acquired rules are useful for modifying the initial rule because of their readability due to the use of ILP.

One drawback of our method is that it tends to overfit the training examples. Future work will include evaluating the acquired rules used to modify the initial classification rules.

References

1. James Cussens. Part-of-speech tagging using progol. S. Džeroski and N. Lavrač, editors, In *Proceedings of the 7th International Workshop on Inductive Logic Programming*, volume 1297, pages 93–108. Springer-Verlag, 1997.
2. Eric Brill. Some advances in transformation-based part of speech tagging. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, volume 1, pages 722–727, 1994.
3. Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.
4. Fumio Mizoguchi and Hayato Ohwada. Constrained relative least general generalization for inducing constraint logic programs. *New Generation Computing*, 13:335–368, 1995.
5. Fumio Mizoguchi and Hayato Ohwada. Using inductive logic programming for constraint acquisition in constraint-based problem solving. In *Proceedings of the 5th International Workshop on Inductive Logic Programming*, pages 297–322, 1995.
6. S. Abney, R. Schapire, and Y. Singer. Boosting applied to tagging and pp attachment. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

Self-generation of Control Rules Using Hierarchical and Nonhierarchical Clustering for Coagulant Control of Water Treatment Plants

Hyeon Bae[1], Sungshin Kim[1], Yejin Kim[2], and Chang-Won Kim[2]

[1] School of Electrical and Computer Engineering, Pusan National University,
609-735 Busan, Korea
{baehyeon, sskim}@pusan.ac.kr
http://icsl.ee.pusan.ac.kr
[2] Dept. of Environmental Engineering, Pusan National University,
609-735 Busan, Korea
{yjkim, cwkim}@pusan.ac.kr

Abstract. In coagulant control of water treatment plants, rule extraction, one of datamining categories, was performed for coagulant control of a water treatment plant. Clustering methods were applied to extract control rules from data. These control rules can be used for fully automation of water treatment plants instead of operator's knowledge for plant control. In this study, statistical indices were used to determine cluster numbers and seed points from hierarchical clustering. These statistical approaches give information about features of clusters, so it can reduce computing cost and increase accuracy of clustering. The proposed algorithm can play an important role in datamining and knowledge discovery.

1 Introduction

The treatment process used depends on the quality and nature of the raw water. Water treatment processes can be simple, as in sedimentation, or may involve complex physicochemical changes, such as coagulation. Several types of chemicals are applied for coagulation. Therefore, it is very important to determine the type and dosage of coagulant [1]. [2].

In the target treatment plant, three coagulants are usually used, such as PAC (Poly Aluminum Chloride), PASS (Poly Aluminum Sulfate Silicate) and PSO-M (Poly Organic Aluminum Magnesium Sulfate). The type and dosage are determined based on a Jar-test, and then the test result is analyzed according to an expert's knowledge. However, at this site, all of the subsystems have been constructed for full automation except a coagulation basin. This causes a bottleneck in fully automatic control of a water treatment plant. Thus, an automatic decision support algorithm is proposed to determine the coagulant type and dosage. In this study, we used the statistical index to determine the cluster number and seed points of fuzzy clustering. The proposed method is easily applied and performance is also adequate for industrial processes.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 371 – 373, 2005.

2 Water Treatment Plant

A water treatment plant involves several processes from the influent water basin to the supplying water line. A water treatment plant should have the capability of purifying water for standard quality and supply to customers even though the quality of the water source gets worse. The principal unit processes of a water treatment plant consist of the following, as shown in Fig. 1.

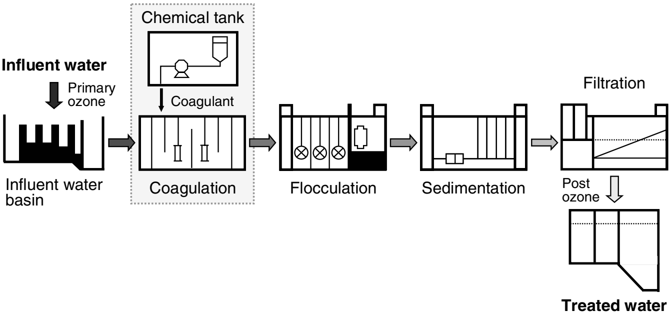


Fig. 1. Basic processes of water purification plant

3 Rule Extraction from Water Data

3.1 Find Initial Points of Fuzzy c-Mean Based on Hierarchical Clustering

Hierarchical methods do not require *a priori* knowledge of the number of clusters of the starting partition. On the other hand, in nonhierarchical methods, the cluster center or initial partition has to be identified before the technique can proceed to cluster observation. The nonhierarchical clustering algorithms, in general, are very sensitive to the initial partition. Therefore, hierarchical and nonhierarchical techniques should be viewed as complementary clustering techniques. In this study, seed points were calculated by clustering results of the hierarchical method. After finding initial partition points, fuzzy c-mean clustering was performed with the calculated seed points.

3.2 Determination of the Number of Clusters Using Statistics Index

In this study, hierarchical clustering was employed in order to obtain cluster information. Using this information, the cluster numbers were determined. The candidate concept for cluster numbers was added, because the result through hierarchical clustering does not support precisely correct information of clusters. Two or three candidate cluster numbers were examined by fuzzy c-mean clustering, which finally determined the number of clusters. Given the cluster solution, the next obvious steps are to evaluate the solution, determine cluster numbers, and extract rules (Fig. 2).

If Temperature is L, Turbidity is L, Alkalinity is H, and **Chl-a is H**, Then Coagulant is PASS

If Temperature is H, **Turbidity is H**, Alkalinity is L, and Chl-a is L, Then Coagulant is PAC

If Temperature is M, Turbidity is M, Alkalinity is M, and Chl-a is M, Then Coagulant is PSO-M

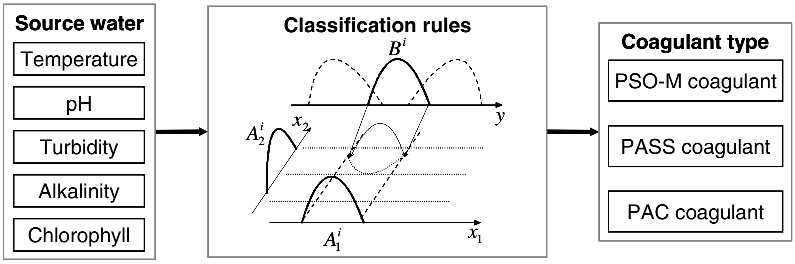


Fig. 2. Coagulant selection rules and the system proposed in this study

4 Conclusions

In this study, the datamining application was accomplished for coagulant control in water treatment plants based on clustering methods and the quality of source water. In the author's previous study, decision tree methods were applied to generate control rules, but they require preliminary knowledge. Alternatively, the proposed clustering method can group inputs corresponding to patterns without preliminary knowledge. Thereby, the clustering method can generate proper rules in rule extraction. Through the proposed algorithm, control rules can be extracted from data to determine the coagulant type automatically.

Acknowledgement

This work was supported by grant No.(R01-2003-000-10714-0) from the Basic Research program of the Korea Science & Engineering Foundation.

References

1. Black, A. P., Hamnah, S. A.: Electrophoretic Studies of Turbidity removal Coagulant with Aluminum Sulfate. J. AWWA **53** (1961) 438
2. Kwang, J. O.: Principle and Application of Physical and Chemical Water Treatment. Gisam publisher (1998) 192-209

A Semantic Enrichment of Data Tables Applied to Food Risk Assessment

Hélène Gagliardi, Ollivier Haemmerlé, Nathalie Pernelle, and Fatiha Saïs

LRI (UMR CNRS 8623 - Université Paris-Sud) / INRIA (Futurs),
Bâtiment 490, F-91405 Orsay Cedex, France
{gag, ollivier, pernelle, sais}@lri.fr

1 Introduction

Our work deals with the automatic construction of domain specific data warehouses. Our application domain concerns microbiological risks in food products. The MIEL++ system [2], implemented during the Sym'Previus project, is a tool based on a database containing experimental and industrial results about the behavior of pathogenic germs in food products. This database is incomplete by nature since the number of possible experiments is potentially infinite. Our work, developed within the e.dot project[1], presents a way of palliating that incompleteness by complementing the database with data automatically extracted from the Web. We propose to query these data through a mediated architecture based on a domain ontology. So, we need to make them compatible with the ontology. In the e.dot project [5], we exclusively focus on documents in Html or Pdf format which contain data tables. Data tables are very common presentation scheme to describe synthetic data in scientific articles. These tables are semantically enriched and we want this enrichment to be as automatic and flexible as possible. Thus, we have defined a Document Type Definition named SML (Semantic Markup Language) which can deal with additional or incomplete information in a semantic relation, ambiguities or possible interpretation errors. In this paper, we present this semantic enrichment step.

2 An Automatic Approach to Enrich Tables Semantically

The data tables which are extracted from the Web are first represented in an XML format using purely syntactic tags : rows and cells. Besides, when it is possible, titles are extracted. We have then to express these data using the vocabulary stored in the ontology. The Sym'Previus ontology contains a taxonomy of 428 terms and a relational schema which describes 25 semantic relations of the domain. In a SML document rows are not represented by cells anymore but by a set of semantic relations between columns.

The semantic enrichment of tables is done in two steps: the first step consists in identifying the semantic relations appearing in the data table. The second step consists in instantiating semantic relations discovered in the table.

[1] Cooperation between INRIA, Paris South University, INRA and Xyleme.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 374–376, 2005.

In order to extract semantic the relations of the table, we first identify the *A-terms*[2] which represent each table column. We look for an A-term which subsumes most of the values. [4] and [6] showed that such techniques give good results when one searches for schema mappings for relational data bases or XML. If the values do not help, we exploit the title of the column. If no A-Term has been found, we associate a generic A-term named *attribute* with the column. Thus we obtain a schema for the table. The schema *tabSch* of the table *Table. 1* is: {*(1,food) (2,attribute) (3,lipid),(4,calorie)*}.

```
<table> <table-title>Nutritional Composition of some food products </table-title >
<column-title> Product </column-title> ... <content> <rowRel additionalAttr=”yes”>
<foodLipid relType=”completeRel”><food attrType=”Normal”>
<ontoVal indMap=”intersection”> whiting Provencale</ontoVal>
<ontoVal indMap=”intersection”> green lemon </ontoVal>
<ontoVal indMap=”intersection”> whiting fillets </ontoVal>
<originalVal> whiting with lemon </originalVal></food>
<lipid attrType=”Normal”> <ontoVal indMap=”notFound”/>
<originalVal> 7.8 g</originalVal> </lipid>
<attribute indMap=”notFound” attrType=”generic”> <ontoVal/>
<originalVal> 100 g</originalVal></attribute> </foodLipid>

<foodAmountLipid relType=”partialNull”> ... <amount attrType=”null”>...
</amount></foodAmountLipid> </rowRel> ... </content> </table>
```

Fig. 1. *SML Representation of the nutritional composition of food products*

Table 1. *Nutritional Composition of some food products*

Products	Qty	Lipids	Calories
whiting with lemon	100 g	7.8 g	92 kcal
ground crab	150 g	11.25 g	192 kcal
chicken	250 g	18.75 g	312 kcal

Then we propose an automatic identification of the semantic relations as flexible as possible. A relation is ***completely represented (CR)*** if each attribute of its signature subsumes or is equal to a distinct A-term of the table schema. A relation is ***partially represented (PR)*** if it is not completely represented and if at least two attributes of its signature subsume or are equal to a distinct A-term of the table schema. In such cases one of the missing attributes may correspond to a constant value which appears in the title of the table. The missing attributes are represented in the SML document by means of an empty tag or by a constant. For example the relation *foodAmountLipid* shown in figure 1 is a **partially represented relation**, where the attribute *amount* is represented by an empty tag. When no relation has been found, a generic relation is generated in order to keep semantic links between values. Fig.1 shows a part of the SML document which is automatically generated from the table shown in Table. 1.

[2] An A-term is a term of the taxonomy that appears at least once as an attribute of a relation signature in the relational schema of the ontology.

Once the relations are extracted, we instantiate them by the values contained in the table. Besides, terms of the ontology are associated with each value when it is possible. The SML formalism allows us to associate several terms that can be found by different mappings procedures. The first one uses simple syntactic criteria. The second one is the unsupervised approach PANKOW [3].

The SML representation of a relation is composed of the set of attributes that appear in the signature of the relation described in the ontology (e.g. *foodLipid*(*food*, *lipid*)). A set of terms represented inside the XML tag *ontoVal* is associated with each value. Thus, three different terms are proposed for *whiting with lemon* : *whiting Provençale*, *green lemon* and *whiting fillets*. The original value is kept inside the XML tag *originalVal* and this value can be shown to the user.

In order to evaluate our approach, we have collected 50 tables from the Web and we have compared the recall, the precision and the F-measure for the different kinds of semantic relations. This result shows that the recall significatively increases when partially identified relations are kept (recall(CR)=0.37 and recall(CR&PR)=0.60) and that the precision do not fall much (0.61 to 0.56).

3 Conclusion

Our method allows one to enrich semantically tables extracted from heterogenous documents found on the Web. The semantic enrichment is completely automatic and is guided by an ontology of the domain. Thus, that processing cannot lead to a perfect and complete enrichment. The SML representation we propose keeps all the possible interpretations, incompletely identified relations and original elements of the context. Contrarily to previous approaches like [1], we cannot base the search for information on a common structure discovered among a set of homogeneous documents.

References

[1] Arvind Arasu and Hector Garcia-Molina, *Extracting structured data from web pages*, Proceedings of the 2003 ACM SIGMOD international conference on Management of data, ACM Press, 2003, pp. 337–348.

[2] Patrice Buche, Juliette Dibie-Barthélemy, Ollivier Haemmerlé, and Mounir Houhou, *Towards flexible querying of xml imprecise data in a dataware house opened on the web*, Flexible Query Answering Systems (FQAS), Springer Verlag, june 2004.

[3] Philipp Cimiano, Siegfried Handschuh, and Steffen Staab, *Towards the self-annotating web*, WWW '04: Proceedings of the 13th international conference on World Wide Web, ACM Press, 2004, pp. 462–471.

[4] AnHai Doan, Ying Lu, Yoonkyong Lee, and Jiawei Han, *Profile-based object matching for information integration*, Intelligent Systems, IEEE **18** (2003), no. 5, 54– 59.

[5] e.dot, *Progress report of the e.dot project*, http://www-rocq.inria.fr/gemo/edot, 2004.

[6] Erhard Rahm and Philip A. Bernstein, *A survey of approaches to automatic schema matching*, The VLDB Journal **10** (2001), no. 4, 334–350.

Knowledge Discovery Through Composited Visualization, Navigation and Retrieval

Wei-Ching Lim and Chien-Sing Lee

Faculty of Information Technology,
Multimedia University, Cyberjaya 63100 Selangor, Malaysia
{lim.wei.ching04, cslee}@mmu.edu.my

Abstract. In large databases, many problems occur when visualizing, navigating and retrieving information from databases. Ontologies help in adding semantics and context to the resources in databases. Hence, this paper presents the OntoVis, an ontological authoring and visualization tool, which emphasizes on the clustering of concepts in Formal Concept Analysis (FCA). The composited visualization, navigation and retrieval of resources will be presented in this paper.

1 Introduction

With the growth of information in databases, it becomes more and more difficult for the users to visualize and search for the interested piece of data, especially in the case of OLAP (Online Analytical Processing) in terms of drill-down analysis. More effective means for visualization, navigation and retrieval is thus needed.

In order to solve the problems mentioned, we developed a visualization tool (OntoVis). OntoVis visualizes concepts based on the updated ontology from OntoShare, which is stored in XML/the database. The OntoShare algorithm (Kiu & Lee, 2005; Lee & Kiu, 2005) allows structural and intentional morphism between ontologies in different electronic databases prior to merging. Discovery of new categories of concepts are automated through SOM-FCA and visualized in the OntoVis.

OntoVis attempts to focus on 3 aspects to efficiently address disorientation, cognitive overload and facilitation of knowledge discovery in databases:

- Automatic generation of trees/lattices from a formal context using Formal Concept Analysis (FCA)
- Ontological retrieval of resources
- Compositional view for navigation

The outline for this paper is as follows: Section 2 automatic generation of the composited visualization and Section 3 retrieval of resources. Section 4 concludes.

2 Automatic Generation of Concept Lattices from Formal Context

The concept lattice (Waiyamai, Taouil & Lakhal, 1997) is used in this paper for ontological structuring, navigation and retrieval. The authoring interface allows the

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 377 – 379, 2005.

designer to indicate how many concepts or topics and attributes to generate. The concepts-attributes relationship forms the context table. Subsequently, the system generates the concept lattice based on the following algorithm:

List all the mathematical combinations of intents that can be created from the formal context, *K:* = (*E, P, R*).

Example: If 3 intents (attributes) are keyed in, then the total combinations will be:
Total combinations: $^3C_0 + {}^3C_1 + {}^3C_2 + {}^3C_3$
* *C* = combination

1. Based on the context table and lists of combinations, create the concepts for the entry marked "Y" (Yes).
2. Construct the concept lattice based on the result from steps 1 and 2.

The automated building of concept lattice in OntoVis is based on the context inputs provided by the user. In addition, OntoVis supports the creation of instances, a feature not provided for in ConExp (Tao, 2003). Editing is done by double clicking on the concept or instances. OntoVis generates a tree or concept lattice depending on the context filled in by the user and does not force a tree to be visualized as a lattice structure.

The grouping of similar concepts leads to a compositional concept lattice, which is a significant difference from ConExp. A compositional view allows the lattice to have different levels of details while ConExp is able to present only a "flat", level-0 concept lattice. The idea of compositing keeps the irrelevant concepts away from the user, thus helping the user to focus on concepts relevant to them. The user can also zoom into deeper levels (instances) and pan across the first level of main concepts (Lee, 2004).

Besides the visualization view, OntoVis shows the formal labeling for the concept nodes. A formal identification of a concept includes all the objects which are explicit (directly contained in it) and implicit (objects which are encapsulated in the concept).

3 Ontological Query and Retrieval of Resources

According to Carpineto and Romano (1995), the extent-intent (object-attribute) pairs are considered complete pairs if

1. the extent is the set of objects described by minimally the properties in the intents
2. the intent is the set of properties shared by all the objects in the extent

The retrieval of data in OntoVis is semantic-based because of the ontological generation of concepts. OntoVis uses the intents of the concept as the query terms for searching. The object(s) retrieved must fulfill all the terms (properties) specified by the query relevant to the object. This means that a query which requires fulfillment of all terms or properties will retrieve the most specific concept in the concept lattice and vice versa. Search generality is increased or decreased as the user traverses up or down the concept lattice using OR or AND operators. This will increase the accuracy of querying and assist the user in knowledge discovery.

Findings from an initial case study show promising results for visualization, navigation and retrieval using the OntoVis (Lee & Lim, 2005).

4 Conclusion

This paper has investigated automatic generation of trees/lattices, ontological retrieval of data and composited navigation. In terms of automatic generation of trees/concept lattices, Formal Concept Analysis (FCA) has been used due to its semantic clustering properties. Besides, retrieval of resources based on the intents/attributes is more efficient and accurate based on semantics. For the navigational part, a composited view is feasible for modeling database structure. A compositional representation allows display of multiple levels of abstraction, allowing a compact view of composite classes leading to object classes and objects (each with its metadata). Future work involves enhancing query and dynamic inclusion strategies.

Acknowledgement

This research is funded by the Malaysian Intensive Research in Priority Areas funding and Multimedia University's internal funding.

References

Brusilovsky, P.:. Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction*, Vol. 6 (2-3) (1996) 87-129.

Kiu, C. C. , Lee, C. S.: Discovering ontological semantics for reuse and sharing of learning objects in a contextual learning environment. *IEEE International Conference on Advanced Learning Technologies (ICALT 2004)*, IEEE Computer Society Press, Kaohsiung Taiwan, July 5-8 2005.

Lee, C. S.: Reuse in modeling instructional design, *World Conference on Educational Multimedia, Hypermedia and Telecommunications.* AACE. Lugano Switzerland, June 21-26 2004.

Lee, C. S., Kiu, C. C.: A concept-based graphical-neural approach to ontological interoperability. *WSEAS Transactions on Computers* (in press).

Lee, C. S., Lim, W. C.: Ontology-based 3-dimensional visualization of composited topics, *World Conference on Educational Multimedia, Hypermedia and Telecommunications,* AACE, Montreal, Canada (2005).

Tao, G.: Using Formal Concept Analysis (FCA) for Ontology Structuring and Building (2003).

Waiyamai, K., Taouil, R., Lakhal, L.: Towards on object database approach for managing concept lattices, *International Conference on Conceptual Modeling/The Entity Relationship Approach* (1997) 299-312.

A Tabu Clustering Method with DHB Operation and Mergence and Partition Operation

Yongguo Liu[1,2,3], Dong Zheng[3], Shiqun Li[3], Libin Wang[3], and Kefei Chen[3]

[1] College of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, P.R. China
[2] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, P.R. China
[3] Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200030, P.R. China

Abstract. A new tabu clustering method called ITCA is developed for the minimum sum of squares clustering problem, where DHB operation and mergence and partition operation are introduced to fine-tune the current solution and create the neighborhood, respectively. Compared with some known clustering methods, ITCA can obtain better performance, which is extensively demonstrated by experimental simulations.

1 ITCA Algorithm

In this article, we focus on the minimum sum of squares clustering problem. Many researches deal with this problem by stochastic optimization methods [1,2,3]. Recently, researchers combine genetic algorithms and K-means algorithm to solve this problem [4,5]. Here, two novel operations, DHB operation and mergence and partition operation, are introduced to fine-tune the current solution and establish the neighborhood, respectively. Therefore, an Improved Tabu Clustering Algorithm (ITCA) is developed. Most procedures of ITCA observe the architecture of tabu search. Here, we pay main attention to these two operations.

***DHB Operation*:** In [6], five iteration methods (DHB, DHF, ABF, AFB, and K-means) are compared. By computer experiments, we choose DHB algorithm as the improvement operation. That is, if object x_i belonging to cluster C_j is reassigned, then the corresponding parameters are updated.

$$\begin{cases} c'_j = (n_j c_j - x_i)/(n_j - 1) \\ c'_k = (n_k c_k + x_i)/(n_k + 1) \\ J' = J - n_j \|x_i - c_j\|^2/(n_j - 1) + n_k \|x_i - c_k\|^2/(n_k + 1) \end{cases} \tag{1}$$

DHB operation is given as follows: Object x_i belonging to cluster C_j is reassigned to cluster C_k, iff $\min \varDelta J_{ik} < \varDelta J_{ij}$, where $i = 1, \ldots, N$, $j, k = 1, \ldots, K$, and $j \neq k$. After all objects are considered, the modified solution is obtained.

***Mergence and Partition Operation*:** We introduce mergence and partition operation to create the neighborhood. Here, we randomly perform one partition

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 380–382, 2005.

and one mergence on solution X_c, keep the number of clusters constant, and form a neighbor. This operation includes four sub operations: 1).*Mergence Cluster Selection*: Here, we introduce proportional selection to determine the cluster to be merged. That is, the closer two clusters to each other, the more possibly one of them is selected as the one to be merged, and vice versa. In the cluster pair, the cluster with sparser structure is the one to be merged. 2).*Partition Cluster Selection*: Like above sub operation, proportional selection is adopted to choose the cluster to be partitioned. That is, the sparser the cluster, the more possibly it is selected as the one to be partitioned, and vice versa. 3).*Cluster Mergence*: Here, objects belonging to the cluster to be merged will be reassigned to their respective nearest clusters. 4).*Cluster Partition*: Here, iteration methods are introduced to divide the cluster to be partitioned into two new clusters. By computer experiments, K-means algorithm is chosen to perform this task.

2 Experiments and Analysis

Figure 1 shows the algorithm equipped with DHB operation can attain the best result more quickly and stably than the other. Figure 2 shows mergence and partition operation is far superior to the probability threshold. Five data sets are considered: German Towns [4], British Towns [4], Data52 [5], Data62 [5], and Vowel [5]. Here, we consider two cases: one is that the number of clusters is variable (German Towns and British Towns); the other is that this parameter is fixed (Data52, Data62, and Vowel). Five methods are considered here: GGA [1], GKA [4], KGA [5], TCA [2], and ITCA. The time complexities of GGA, GKA, KGA, and TCA are $O(GPKmN)$, $O(GPmN^2)$, $O(GPKmN)$, and $O(GN_tmN)$, respectively. For ITCA, its time complexity is equal to $O(GN_tm(KN'+N)))$. In many cases, $KN' < N$, the cost of ITCA is close to that of TCA. If the mergence is performed before the partition, the complexity of ITCA will be similar to that of TCA and far lower than those of GGA, GKA, and KGA. The average (Avg) and standard deviation (SD) values of the clustering results and their success

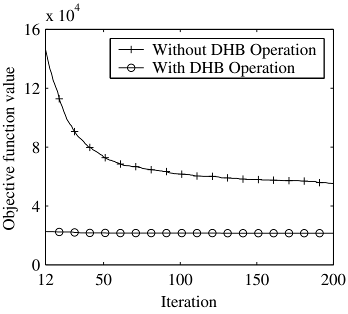


Fig. 1. Comparison of different modes for improving solution X_c

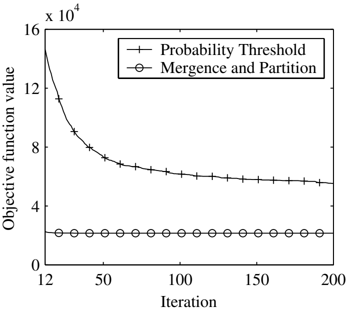


Fig. 2. Comparison of different modes for creating the neighborhood

Table 1. Comparison of the clustering resutls of five methods for different data sets

		GGA	GKA	KGA	TCA	ITCA
GT4C	Avg	50872.12	49600.59	49600.59	78343.36	49600.59
	SD	5016.98	0.00	0	4348.65	0
	SR(%)	0	100	100	0	100
GT6C	Avg	31988.74	30806.99	30535.39	63465.08	30535.39
	SD	1186.72	805.12	0	4374.93	0
	SR(%)	0	85	100	0	100
GT8C	Avg	23833.86	21787.32	21566.13	53091.34	21483.02
	SD	1236.95	347.12	241.77	3582.64	0
	SR(%)	0	5	55	0	100
GT10C	Avg	18729.89	17180.32	16502.70	48505.45	16357.39
	SD	795.36	381.71	101.64	3907.92	85.61
	SR(%)	0	0	15	0	75
BT4C	Avg	193.11	181.11	180.91	218.96	180.91
	SD	8.94	0.45	0	5.60	0
	SR(%)	0	80	100	0	100
BT6C	Avg	173.20	142.84	142.06	183.03	141.46
	SD	24.19	1.96	0.96	6.85	0
	SR(%)	0	55	65	0	100
BT8C	Avg	156.04	116.35	114.40	162.36	113.58
	SD	34.95	2.41	0.98	8.27	0.15
	SR(%)	0	20	25	0	80
BT10C	Avg	163.48	95.71	93.50	145.13	92.70
	SD	46.93	2.24	1.17	6.95	0.02
	SR(%)	0	0	25	0	60
Data52	Avg	492.34	488.08	488.02	2666.24	488.02
	SD	6.12	0.13	0	52.87	0
	SR(%)	0	50	100	0	100
Data62	Avg	806.43	543.17	543.17	19592.24	543.17
	SD	353.36	0	0	296.33	0
	SR(%)	0	100	100	0	100
Vowel	Avg	33970758.08	31017497.96	30690310.02	250440977.65	30686385.75
	SD	2444084.44	490397.18	11187.13	2462485.38	351.67
	SR(%)	0	0	20	0	85

rates (SR) are compared as shown in Table 1. GGA is better than TCA. KGA is slightly better than GKA and they are both better than GGA and TCA. It is seen that ITCA is the best among five algorithms and can achieve the best results more stably than other methods especially in complicated ones.

Acknowledgements. This research is supported in part by the National Natural Science Foundation of China (Grants 60473020, 60273049, 90104005) and State Key Laboratory for Novel Software Technology at Nanjing University.

References

1. Hall, L.O., Ozyurt, B., Bezdek, J.C.: Clustering with a genetically optimized approach. IEEE Trans Evol Comput. **3** (1999) 103–112
2. Al-sultan, K.S.: A tabu search approach to the clustering problem. Pattern Recognit. **28** (1995) 1443–1451
3. Bandyopadhyay, S., Maulik, U., Pakhira, M.K.: Clustering using simulated annealing with probabilisitc redistribution. Int J Pattern Recognit Artif Intell. **15** (2001) 269–285
4. Krishna, K., Murty, M.N.: Genetic K-means algorithm. IEEE Trans Syst Man Cybern, Part B-Cybern. **29** (1999) 433–439
5. Bandyopadhyay, S., Maulik, U.: An evolutionary technique based on K-means algorithm for optimal clustering in R^N. Inf Sci. **146** (2002) 221–237
6. Zhang, Q.W., Boyle, R.D.: A new clustering algorithm with multiple runs of iterative procedures. Pattern Recognit. **24** (1991) 835–848

Discovering User Preferences by Using Time Entries in Click-Through Data to Improve Search Engine Results

Parthasarathy Ramachandran

Indian Institute of Science, Bangalore KA 560 012, India
parthar@mgmt.iisc.ernet.in

Abstract. The search engine log files have been used to gather direct user feedback on the relevancy of the documents presented in the results page. Typically the relative position of the clicks gathered from the log files is used a proxy for the direct user feedback. In this paper we identify reasons for the incompleteness of the relative position of clicks for deciphering the user preferences. Hence, we propose the use of time spent by the user in reading through the document as indicative of user preference for a document with respect to a query. Also, we identify the issues involved in using the time measure and propose means to address them.

The unstructured and noisy data on the web poses serious challenges to web search engines. The problem of retrieval quality of the search engines is further augmented by systematic spamming aimed at improving the page ranking practised by the page owners. The most popular technique called the Page-Rank for producing an importance ranking of the pages on the web took advantage of the link structure of the web [1]. Another popular approach is to include anchor-text analysis in the scoring function [2]. These techniques can be classified as *indirect* approaches of evaluating web-page quality with respect to a query. A more *direct* approach to the web-page quality problem was necessitated due to the fact that the indirect methods are very susceptible to spamming. However users are very reluctant to give direct feedback on the web-page quality. User feedback can be implicitly collected, albeit partially, using web search engine logs.

Given that the link to document d_a was clicked before document d_b, it represents the relative quality judgement of the user on the documents with respect to the query. Joachim's proposed a method that used the click-through data with the relative quality judgement's to train a retrieval function. This method uses the relative position of the clicks to deduce the user preference with respect to the query [3]. Another more recent model for using the click-through data uses the co-visited relationship of the web-pages across queries [5]. The co-visited principle means that if two web-pages are visited by users with the same query, then the two web-pages are co-visited. Using this similarity concept the authors developed an iterative algorithm that measures similarities between queries and web-pages and used them in rearranging the search engine results.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 383–385, 2005.

The relative quality judgement based on the relative position of the clicks in the click-through data is incomplete for the following reasons:

- Since the users only review a limited search result pages, the pages down in ranking is less likely to be clicked on independent of the pages relevance for the query. In fact, analysis has shown that for 85% of the queries only the first result screen is reviewed by the user [4].
- Since only a limited number of links are presented in a page the relative quality judgement can at best be made only among the links presented in that page.
- Even with in a single pages, users rarely review all the links presented before selecting the pages to view. Hence it is very unlikely that links that are down the list even with in a single search result page will be viewed ahead of pages that are listed above them.
- The text accompanying the links to web-pages in search results pages are often short and incomplete. Hence the users decision is based on incomplete information.

When a user visits a page, the time spent in reading through its contents cab be used as a proxy for the user feedback on the relevancy of a document to the posed query, assuming that the user's attention is not distracted to other topics. Hence we propose to use the time measure, and for that purpose, we identify three major issues in using the time measure namely, last visited page dilemma, inter-session and intra-session issues.

Last Visited Page Dilemma

Under ideal conditions, if a user with a query visited document d_a at time t_a and and the very next visited document document d_b at time t_b, then the time spent by the user reading through the web-page d_a is given by $\tau_a = t_b - t_a$. This measure can be computed for all but the last page visited by the user. This can be construed as either the user found perfect page answering the posed query or the user just stopped looking further even without finding a page that matched the posed query. It would be difficult to resolve this dilemma without a direct feedback from the user.

Though it would be ideal to have the user feedback in resolving the *last visited page dilemma* (LVPD) described above, the dilemma could be addressed to some extent by using the co-visited relationships of web-pages. This dilemma could be addressed by looking at the behaviour of users with other queries that are similar to the one posed by the current user. Let the set of all queries related to query q be given by $Q(q)$. The web-pages that are returned by the search engine for query q is given by $D(q)$. The web-pages that are clicked by users with queries that are related to each other are defined to have a co-visited relationship, i.e., given a query q, the co-visited pages are $d \in D(q'), \forall q' \in Q(q)$ and it is represented by $C(q)$.

If for all the related queries τ is unknown for a specific page d, then it is quite possible that this document is the most relevant page for these queries. However, knowing how we browse the web, it is quite unlikely that τ will be unknown for all related queries, or in other words, it is quite unlikely that a specific web-page will be the last page visited by all the users with related queries. Hence hopefully there will be some users who have visited other pages after visiting the page under consideration. This will provide us with information for assessing the relevance of the document for a query.

Inter-Session Issues

Different users browse through the pages at varying speeds. Hence, the time spent by a user reading through a page cannot be directly used as a measure of documents relevancy. It is important to normalise the time spent by the user in a web-page before using it. The normalisation procedure should maintain the relative relevance judgement specified by the time spent the user browsing through the web-pages. Further across sessions that are generated by different users the relative relevance judgement's should be comparable.

Intra-Session Issues

The computing platform, Internet connection speeds and the reading speeds vary among the users. Hence a direct comparison of the times as computed from the log files will be grossly misleading. The normalised times will be able to adjust for the varying reading speeds of the users. However, the varying connection speeds and platform processes could still influence the time measures in the log file. One solution to this problem would be consider a measure like the time spent per kilobyte of data accessed. Given the multiple modes of document presentation on the web like pdf, doc, etc., which can only be viewed after completely downloading the entire document, a measure like time spent per kilobyte of data becomes crucial.

Though it seems that time will be good proxy for direct user feedback, future work needs to implement it to prove its efficacy.

References

1. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web Conference*, pages 107–117, 1998.
2. Nick Craswell, David Hawking, and Stephen Robertson. Effective site finding using link anchor information. In *ACM SIGIR*, pages 250–257, New Orleans, 2001.
3. Thorsten Joachims. Optimizing search engines using clickthrough data. In *SIGKDD*, pages 133–142, Alberta, Canada, 2002.
4. C. Silverstein, M. R. Henzinger, J. Marais, and M. Moricz. Analysis of a very large altavista query log. *SIGIR Forum*, 33:6–12, 1999.
5. Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Yong Yu, Wei-Ying Ma, WenSi Xi, and WeiGuo Fan. Optimizing web search using web clickthrough data. In *CIKM*, pages 118–126, Washington DC, 2004.

Network Boosting for BCI Applications

Shijun Wang, Zhonglin Lin, and Changshui Zhang

Department of Automation, Tsinghua University, Beijing 100084, China
{wsj02, linzl02}@mails.tsinghua.edu.cn, zcs@mail.tsinghua.edu.cn

Abstract. Network Boosting is an ensemble learning method which combines learners together based on a network and can learn the target hypothesis asymptotically. We apply the approach to analyze data from the P300 speller paradigm. The result on the Data set II of BCI (Brain-computer interface) competition III shows that Network Boosting achieves higher classification accuracy than logistic regression, SVM, Bagging and AdaBoost.

BCI (Brain-computer interface) is a direct cybernetic link between a mind and a computer which does not depend on the brain's normal output pathways of peripheral nerves and muscles [1]. Most BCIs make use of mental tasks that lead to distinguishable *electroencephalogram* (EEG) signals of two or more classes. P300 potentials provide a means of detecting user's intentions concerning the choice of objects within a visual field. Farwell and Donchin [2] first introduced P300 potentials into BCI, and proposed P300 speller paradigm [3]. In P300 speller paradigm the key task is to detect presence or absence of the P300 component in noisy EEG signals accurately and fast.

Network Boosting (NB) [4] is a new ensemble learning method which combines classifiers on the basis of a network. Theoretic analysis based on the game theory shows that the algorithm can learn the target hypothesis asymptotically. NB is more suitable than other ensemble learning methods for noisy data and distributed application. In this paper We utilize NB for classifying *electroencephalogram* (EEG)-signals to detect absence or presence of the P300 component in EEG event related potentials.

The basic idea of NB is that through the cooperation between classifiers, we expect the learned classifier ensemble has high accuracy as well as high resistance to the noise. The idea comes from our recent research [5] on complex network [6]. In order to facilitate the cooperation of classifiers, a network topology is introduced which serves as a communication structure between them.

Fig. 1 shows the NB algorithm. Assume there are K nodes in the network and the training round is T. In the learning phase, given training set $Z = \langle (x_1, y_1), (x_2, y_2), ..., (x_l, y_l) \rangle$, each classifier on the classifier network is provided with the same training instances and maintains a weight record $w_{k,t}(i)$ for $k = 1, ..., K$, $t = 1, ..., T$, $i = 1, ..., l$ of the instances respectively. Then the classifier in the classifier network is built by the training set sampled from the training data according to the weights record. After that, the weights of the instances of

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 386–388, 2005.

every node are updated according to the classification results of the node and its predecessors. The classifier network is trained T rounds in such way.

We compare 5 algorithms on data set II of BCI Competition III [7]. In P300 speller paradigm, the user faces a 6 × 6 matrix of letters. The user's task is to focus attention on characters in a word that was prescribed by the investigator (i.e., one letter at a time). All rows and columns of this matrix are successively and randomly intensified at a rate of 5.7Hz. Two out of 12 intensifications of rows or columns contains the desired letter (i.e., one particular row and one particular column). The responses evoked by these infrequent stimuli (i.e., the 2 out of 12 stimuli that did contain the desired letter) are different from those evoked by the stimuli that do not contain the desired letter.

```
Algorithm Network Boosting

Input:  Examples $Z = \langle (x_1, y_1), (x_2, y_2), ..., (x_l, y_l) \rangle$
        Directed Network $N$
        Training rounds $T$
        Sampling parameter $\rho$
        Weight update parameter $\beta$

Initialize:  $w_{k,1}(x_i) = 1$ for all sample $i = 1, ..., l$ and node $k = 1, ..., K$

Do for:  1. Generate a replicate training set $T_{k,t}$ of size $l\rho$, by weighted
            sub-sampling with replacement from training set $Z$ for $k = 1, 2, ..., K$.
         2. Train the classifier (node) $C_k$ in the classifier network with respect
            to the weighted training set $T_{k,t}$ and obtain hypothesis
            $h_{k,t} : x \mapsto \{-1, +1\}$ for $k = 1, ..., K$ .
         3. Update the weight of instance $i$ of node $k$ :

            $w_{k,t+1}(i) = w_{k,t}(i)\,\beta^{I(h_{k,t}(x_i)=y_i)+\sum_n I(h_{n,t}(x_i)=y_i)} / Z_{k,t},$    (1)

            where node $n$ is predecessor of node $i$. $I$ is indication function and $Z_{k,t}$
            is a normalization constant, such that $\sum_{i=1}^{l} w_{k,t+1}(x_i) = 1$.

Output:  Final hypothesis by majority voting using the learned hypotheses
         $h_{k,t} : x \mapsto \{-1, +1\}$ for $k = 1, ..., K$ and $t = 1, ..., T$.
```

Fig. 1. Algorithm Network Boosting

The data set comes from two subjects' experiments. For evaluation, we only use the labeled training set. In preprocessing, we find that epoch 11, 62 and 63 of subject A have much larger amplitude than others and we treat them as outliers and discard them. So there are 82 epochs from subject A and 85 epochs from subject B. Then the preprocessing are performed as following: All data are band-pass filtered between 0.5-15Hz; The No. 34, 11, 51, 62, 9, 13, 49, 53, 56 and 60 channels are selected; Signals (lasting 900ms from stimulus) from above channels are concatenated, and then down-sampled to 1/8.

All the algorithms are used to classify single-trial signal. If a signal is judged to have P300 potential, the corresponding code's score is incremented. After 15 classifications for each code, the two codes which gain the highest score gives the target character. We divide epochs as 4 folds, taking 3 as training set and the remaining one test, for subject A and B respectively. After 4 repetitions, we predict all characters. Table.1 gives the error rates by algorithms: Logistic Regression (LR), Support Vector Maching (SVM), Bagging, AdaBoost and NB. Logistic Regression as base classifier is used in all the ensemble learning methods (SVM, Logistic Regression, Bagging and AdaBoost are implemented using WEKA [8].) For Bagging and AdaBoost, 100 base classifiers were used. For NB, $NB(100, 10, 1/3, 0.7)$ and directed random network with connection probability 0.03 (for each directed link) is used.

Table 1. Comparisons of 5 methods on Data set II of BCI competition III

Name	Logistic Regression	SVM	Bagging	AdaBoost	Network Boosting
Subject A (82)	10.98%	10.98%	14.63%	9.76%	6.10%
Subject B (85)	7.06%	17.65%	11.76%	10.59%	5.88%

The comparison results show that NB achieves higher accuracy than others and is more robust than other ensemble learning methods. In the present work, all the data from different channel are combined together as the training data of classifiers. What if we apply one classifier for one channel in the NB algorithm and combine the final results together? It may be a way of future research.

References

[1] Wolpaw, J., et al.: Brain-computer interfaces for communication and control. Clin. Neurophysiol., vol 113. (2002) 767-791

[2] Farwell, L.A., and Donchin, E.: Talking off the top of your head: toward a mental prothesis utilizing event-related brain potentials. Electroenceph. Clin. Neurophysiol, vol. 70. (1988) 510-523

[3] Donchin, E., et al.: The mental prosthesis: assessing the speed of a P300-based brain-computer interface. IEEE Trans. Rehabi. Eng., vol. 8. (2000) 174-179

[4] Wang, S.J., Zhang, C.S.: Network game and boosting. In The 16th European Conference on Machine Learning, (2005)

[5] Wang, S.J., Zhang, C.S: Weighted competition scale-free network. Phys. Rev. E **70**, 066127 (2004)

[6] Albert, R. and Barabási, A.: Statistical mechanics of complex networks. Reviews of Modern Physics, 74 (1). (2002) 47-97

[7] http://ida.first.fraunhofer.de/projects/bci/competition_iii/

[8] Witten, I.H. and Frank, E.: Data Mining: Practical machine learning tools with Java implementations, Morgan Kaufmann, San Francisco, (2000)

Rule-Based FCM: A Relational Mapping Model

Ying Yang[1,2], Tao-shen Li[2], and Jia-jin Le[1]

[1] Institute of Computer & Information Science, DongHua University,
200051, Shanghai, P.R.China
[2] College of Computer & Information engineering, Guangxi University,
530004, Guangxi , P.R.China
yingy2004@126.com

Abstract. Rule-Based Fuzzy Cognitive Map (RBFCM) is proposed as an evolution of Fuzzy Causal Maps (FCM) to allow more complete and complex representation of cognition so that relations other than monotonic causality are made possible. This paper shows how RBFCM can be viewed in the context of relation algebra, and proposes a novel model for representing and reasoning causal knowledge relation. The mapping model and rules are introduced to infer three kinds of causal relations that FCM can't support. Capability analysis shows that our model is much better than FCM in emulating real world.

1 Motivation

Fuzzy Conceptual Maps have become an important means for drawing a graphical representation of a system , and connecting the state concepts (variables) in the system by links that symbolize cause and effect relations, and have been used in simulating process, forecasting or decision support, etc. Though FCMs have many desirable properties, they have some major limitations [4]. For example, .FCMs can't provide the inference of sequential relations or time-delay causal relations because all the interaction of FCMs' concepts is synchronous, and can't provide the inference of conditional probabilistic causal relations. Their inference results in some intelligent systems are usually distorted.

Some authors have tried to extend FCMs to include time, and they developed systems such as "Extended FCMs" (Hagiwara [2]) and rule-based FCMs (Carvalho [3]).But they can't support conditional probabilistic causal relations. Neural Cognitive Map (NCM [5]) are presented to solve complex causal relations, but NCM needs much training data that are difficult to be obtained in some intelligent systems, and time-delay causal relations as well as sequential relations are difficult to be found by neural networks.

Our model proposes a novel model for representing and reasoning causal knowledge relation. The mapping model and rules are introduced to infer three kinds of causal relations including sequential relation, time-delay causal relations, and conditional probabilistic causal relations that FCM can't support.

2 The Mathematical Model

In our model, causal knowledge is in the form of concepts, relations, directional connections and weights. Fig.1 describes the cause-effect relation mapping about terror

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 389 – 391, 2005.

events represented by RBFCM. The hostage, explosion and casualty are the subsequences of terrorists, and the terrorists are the subsequence of the foreign policy and the striking power. The foreign policy has not an immediate effect because it needs days or months to make a full impact on terrorists. The striking power also needs hours or days to make a full impact on terrorists. Our model is the map that can represent and infer more complex and complete casual knowledge than FCM.

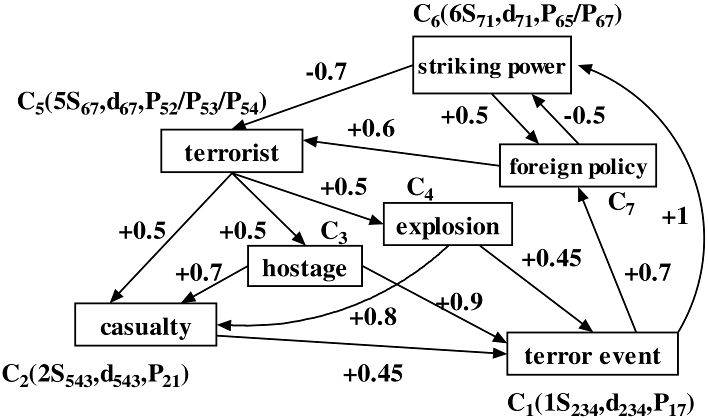


Fig. 1. The cause-effect relation mapping terror events represented by RBFCM

Definition. Let S_{ci} and S_{cj} be the state values of concept C_i and C_j, $f(x)$ be the reasoning function of C_j. R_{ij} and w_{ij} are the relation type and the weight from C_i to C_j respectively, and are the elements of the relation matrix $\boldsymbol{R}$ and the adjacency matrix A. The mapping model can be determined by the following operations of **Rule 1- 3**:

1. If there exist conditional probabilistic causal relations from different concepts C_i to C_j, $f(x)$ is a computing function of all the concepts C_i occurrences leading to the increase / decrease probability of concept C_j. Here $f(x) = \tanh(x) \in [-1,1]$. **2.** If there exists time-delay causal relation from C_i *t*o C_j, then reserve the primary value of C_i during the time delay, and all values of the i^{th} row are zero in the adjacency matrix $\boldsymbol{A}$, then set $S_{ci}(t)$ equal to the original value of C_i. Here $f(x) = 1/(1+e^{-x}) \in [0,1]$. **3.** The (i-1)th sequential relation should be reasoned before the reasoning of the ith sequential relation. if the concept value interval is [-1, 0], Then $f(x) = 1/(1-e^{x})$.

The effect concept's state value at time (*t*+1) should partly depend on its own state value at time *t*. ϕ and φ are allotted coefficient. $\phi + \varphi = 1$. The computing of effect concept's state value is as follows: $S_{cj}(t+1) = f(\phi\sum_{i=1}^{n} w_{ij}S_{ci}(t) + \varphi\sum_{j=1}^{n} w_{ij}S_{cj}(t))$.

The relation matrix $\boldsymbol{R}$ and the adjacency matrix $\boldsymbol{A}$ describe the relation types and the weights between concepts of directional connections respectively. And all of their interactions among concepts, relations, directional connections and weights compose a dynamic network. In Fig.1, the cause-effect causal relation and the conditional probabilistic causal relation are denoted as R_{ce} and R_{cp} respectively. If there does not exist causal relation between concepts, it is denoted as *N* in ***R***. The m^{th} subsequence is denoted as *mS* and the time-delay causal relation is denoted as d_n. For example, ***R2*** (2, 1) = ($2S$, d_{543}) represents that there exists first sequential relation and the time-delay causal relation from C_2 to C_1.. The denotation C_5 ($5S_{67}$, d_{67}, P_{52}/ P_{53}/ P_{54}) in Fig.1 represents that there exists first sequential relation and the time-delay causal relation from C_6 and C_7 to C_5, and there also exists conditional probabilistic causal relation from C_5 to C_2, from C_5 to C_3 and from C_5 to C_4 .

$$R=\begin{pmatrix} N & N & R_{ce} & R_{cp} & N & N \\ R_{ce} & R_{cp} & R_{cp} & (6S,d_{71}) & N & N \\ R_{ce} & N & (5S,d_{67}) & R_{cp} & R_{cp} & R_{cp} \\ N & R_{ce} & R_{cp} & N & N & N \\ N & R_{ce} & R_{cp} & N & N & N \\ R_{ce} & R_{ce} & R_{ce} & (2S,d_{543}) & N & R_{cp} \\ R_{ce} & R_{ce} & R_{ce} & (1S,d_{432}) & N & R_{cp} \end{pmatrix} \qquad A=\begin{pmatrix} 0 & 0 & +0.7 & +0.6 & 0 & 0 \\ +1 & -0.7 & +0.5 & +0.7 & 0 & 0 \\ +0.6 & 0 & -0.7 & +0.5 & +0.5 & +0.5 \\ 0 & +0.45 & +0.5 & 0 & 0 & 0 \\ 0 & +0.7 & +0.5 & 0 & 0 & 0 \\ +0.5 & +0.7 & +0.8 & +0.45 & 0 & +0.45 \\ +0.45 & +0.9 & +0.45 & +0.7 & 0 & +1 \end{pmatrix}$$

3 Conclusions and Future Work

In this paper, RBFCM represent a very promising inference structure that is able to capture the causal reasoning processing present in most human decision making activities. We present our formal definitions and theoretical results for the analysis of the inference mechanisms of RBFCM. Although in real-world applications, FCM can be extremely complex, we can regularly divide a given FCM into basic FCM modules. The ongoing work is to use the mapping model to share and understand causal knowledge in Knowledge Grid environment.

References

1. Osei-Bryson, K., Generating consistent subjective estimates of the causal relationships in fuzzy cognitive maps. Computers & Operations Research 2004, 31: 1165-1175
2. M. Hagiwara. Extended Fuzzy Cognitive Maps. In: the proc of IEEE International Conference on Fuzzy System FUZZ-IEEE, San Diego, 2002, 795-801.
3. Carvalho, J.P., J.A.B. Tomé, . Rule Based Fuzzy Cognitive Maps – Expressing Time in
4. Qualitative System Dynamics, Proceedings of the FUZZ-IEEE2001
5. Özesmi, U., S.L. Özesmi, Ecological models based on people's knowledge: a multi-step fuzzy cognitive mapping approach, Ecological Modelling 2004 176 (1-2): 43-64
6. T. Obata, Neural Cognitive Maps. http://citeseer.ist.psu.edu/.

Effective Classifier Pruning with Rule Information

Xiaolong Zhang[1,2], Mingjian Luo[1], and Daoying Pi[2]

[1] School of Computer Science and Technology,
Wuhan University of Science and Technology, Wuhan 430081 P.R. China
{xiaolong.zhang, mingjian.luo}@mail.wust.edu.cn
[2] Dept. of Control Science and Engineering,
Zhejiang University, Hangzhou 310027 P.R. China
dypi@iipc.zju.edu.cn

Abstract. This paper presents an algorithm to prune a tree classifier with a set of rules which are converted from a C4.5 classifier, where rule information is used as a pruning criterion. Rule information measures the goodness of a rule when discriminating labeled instances. Empirical results demonstrate that the proposed pruning algorithm has high predictive accuracy[1].

1 Introduction

Decision tree pruning is a kind of method to improve predict accuracy and avoid overfitting. There are some approaches previously proposed in this area. Minimum description length pruning [1] and minimal cost complexity pruning [2] generate a sequence of pruned trees and later select better one from them. It should be noted these methods prune a decision tree in a direct way, pruning nodes among the paths from a tree node to a tree leaf. It cannot delete a condition if the reorganization of the tree fails. This paper proposes a tree pruning method, pruning a tree with a rule set. The rule set is converted from a decision tree [3], where rule information proposed in [4] is used as pruning criteria. It prunes a tree in an indirect way. The tree pruning with a set of rules has some advantages over those done by a direct way. It can avoid the over pruning caused in the previous pruning methods especially when the training set is small.

Rule information [4] is used to calculate relationship between antecedent (conditions) and consequent (prediction) of a rule. Rule information is suited to describe the relationship among the conditions and the prediction. Moreover, a rule belief is employed to identify a rule if the rule is necessary to be pruned before a rule pruning starts. Empirical tests and comparisons show that our algorithm outperforms C4.5 in predictive accuracy.

[1] The work is supported by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, and the Project (No.2004D006) from Hubei Provincial Department of Education, P. R. China.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 392–395, 2005.

2 Rule Information and Rule Belief

A decision tree can be converted to a set of rules. The rules with IF-THEN form are used to prune a tree. A rule is expressed with $\sum_{i=i}^{n}(A_1 = v_{ij}) \rightarrow (C = c_m)$, where $A_i = v_{ij}$ is a condition of the rule and $C = c_m$ the prediction of the rule, n is the number of the attributes; v_{ij} a value of the attribute A_i, c_m a class of the prediction C (assume m is among $1 \cdots k$, k denotes the number of the class values of C).

Rule information describes the mutual relationship for a rule between its conditions and its prediction. Rule information [4] is defined as

$$I(R) = \log_2 \frac{P(C = c_m | \prod_{i=1}^{n}(A_i = v_{ij}))}{P(C = c_m)}$$

where $P(C = c_m|(A_i = v_{ij}))$ is the proportion of attribute A_i with value v_{ij} under $C = c_m$, $P(C = c_m)$ is the proportion of class C with value c_m in the training set.

As the definition of $I(R)$, a rule with a larger value denotes the relationship between the condition part and the prediction part in a rule is tighter. If $C = c_m$ and all the examples in the training data are covered, $I(R)$ is the maximum $-\log_2 P(C = c_m)$. If the prediction of a rule cannot derive from inadequate conditions, its rule information is probably negative. Our algorithm only deals with the rules whose rule information ranges from 0 to $-\log_2 P(C = c_m)$. To identify a rule if it can be a candidate to be pruned, we define a concept "rule belief": $B(R) = \frac{I(R)}{-\log_2 P(C=c_m)}$. B(R) is a normalization of $I(R)$ with $-\log_2 P(C = c_m)$, which can be directly used to identify whether a rule should be deleted before pruning.

Both rule information and rule belief are used in our rule pruning algorithm. Rule information is used as a pruning criterion. Given a rule r, every step in rule pruning of r should not decrease the rule information $I(r)$. Rule belief is used to identify whether a rule should be directly deleted without entering rule pruning process.

3 Pruning Algorithm and Experiments

Our tree growing process is similar to that of C4.5. A built decision tree is then converted to a rule set. The algorithm mainly consists of: 1) Calculation of rule information and rule belief for each rule; 2) Deletion of the rules whose rule belief is less than a given threshold δ; 3) For each rule, any condition of the rule can be removed once it does not worsen the rule information of the rule. In every rule pruning, so that the increment of rule information should be greater than a value ϵ (given according to the noisy rate in the training data). For every condition of the rule, the algorithm tries to find the most effective pruning, where the increment of the rule information is maximum. The pruned rule with the most effective pruning is inserted into the pruned rule set R^p.

The experiments have extensively been evaluated with 4 data sets (Connect-4, Breast-cancer, Iris Plants, Credit-screening) selected from the well-known UCI data repository [5]. The predictive accuracy on their testing set is calculated. The results demonstrate our rule pruning algorithm outperforms C4.5 in predictive accuracy. For each data set, we have carried out several training and testing respectively. In case of Connect-4 (10-600-100), 10 training-testing tunes are performed; the first training set includes 600 examples; and each of the next training set is increased with 100 examples which are randomly selected from its data source. For Breast-cancer, Iris Plants, and Credit-screening, the above number is (10-50-50), (10-50-10), (9-300-30), respectively. Predictive accuracy is calculated with every training-testing tune by applying the pruned tree on the original data source except the training examples.

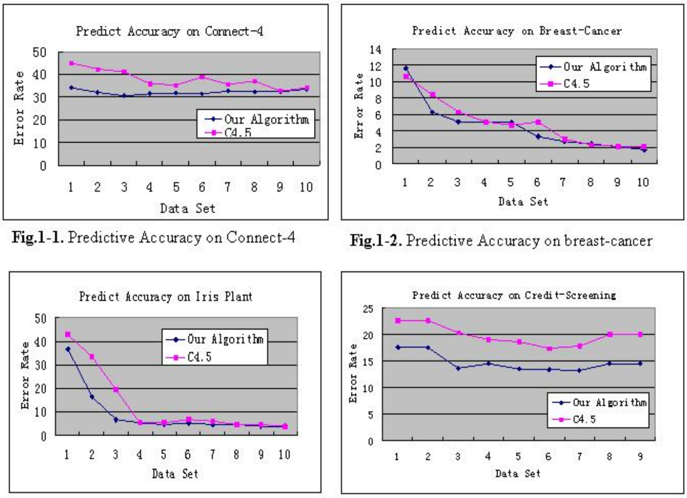


Fig.1-1. Predictive Accuracy on Connect-4

Fig.1-2. Predictive Accuracy on breast-cancer

Fig.1-3. Predictive Accuracy on Iris Plant

Fig.1-4. Predictive Accuracy on credit-screening

Fig. 1. Predict accuracy comparison

The results (see Fig. 1) show the predictive accuracy of these four conducted domains. Fig.1-1 shows both C4.5 and our algorithm have high error ratios (more than 30%) on Connect-4. The predictive accuracy obtained on Breast-Cancer (Fig. 1-2) and Iris Plant (Fig. 1-3) is reasonable, where the error ratios decrease when the training examples increase. However, for Credit-screening (Fig. 1-4), the increasing examples seem to be not improving the predictive accuracy. Both C4.5 and our algorithm have similar accuracy curves on these 4 domains. Our algorithm performs better than C4.5 on these 4 domains, and learns more accurate classifiers.

4 Conclusion

The proposed tree pruning method with rule information and rule belief performs better than C4.5 in the predictive accuracy. One of the future works is to explore the model optimization with the decision tree.

References

1. Mehta M., Rissanen J., Agrawal R. MDL-Based Decision Tree Pruning. *Proceedings of the First International Conference on KDD*, 216-221, 1995.
2. Breiman L., et al. Classification and Regression Trees. *Wadsworth & Brooks Press*, 1984.
3. Quinlan J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.
4. Hu D., Li H.X. Rule Mining and Rule Reducing Based on the Information of Rules. *Pattern Recognition and Artificial Intelligence*, 17(1), 2004.
5. Blake C., Merz C. UCI Repository of Machine Learning Databases. *Dept. of Information and Computer Science, University of California.*

Text Mining for Clinical Chinese Herbal Medical Knowledge Discovery

Xuezhong Zhou[1], Baoyan Liu[1], and Zhaohui Wu[2]

[1] China Academy of Traditional Chinese Medicine, Beijing, 100700, P.R. China
{zxz, liuby}@mail.cintcm.ac.cn
[2] College of Computer Science, Zhejiang Univeristy, Hangzhou, 310027, P.R. China
wzh@cs.zju.edu.cn

Abstract. Chinese herbal medicine has been an effective therapy for healthcare and disease treatment. Large amount of TCM literature data have been curated in the last ten years, most of which is about the TCM clinical researches with herbal medicine. This paper develops text mining system named MeDisco/3T to extract the clinical Chinese medical formula data from literature, and discover the combination knowledge of herbal medicine by frequent itemset analysis. Over 18,000 clinical Chinese medical formula are acquired, furthermore, significant frequent herbal medicine pairs and the family combination rule of herbal medicine have primary been studied.

1 Introduction

Recently, text mining has attracted great attention in the biomedical research community [1,2,3] due to the large amount of literature and TextBases (e.g. Medline) have been accumulated in the biomedical fields.

Traditional Chinese Medicine (TCM) has been a successful approach for Chinese health practice since several thousand years ago. It is significant to study the compositional rule of Chinese Herbal Medicine (CHM) since CHM has been a novel basis of new drug development. The TCM bibliographic database, which contains over one half million records from 900 biomedical journals published in China since 1984[1]. This paper aims to discover knowledge from TCM literature with regard to clinical CMF (Chinese Medical Formula) component CHM combination. We follow the approach suggested in [4] to extract the structured objective information and then apply the traditional data mining algorithms. We develop a text mining system called MeDisco/3T(**Me**dical **Disco**ver for **T**raditional **T**reatment in**T**elligence) to mine the CHM knowledge from TCM literature. Firstly, MeDisco/3T extracts structured CMF information (e.g. CMF name, CHM components and efficacy description) from literature based on bootstrapping method [5]. Secondly, it uses frequent itemset algorithm to analyze the data.

2 MeDisco/3T Text Mining System

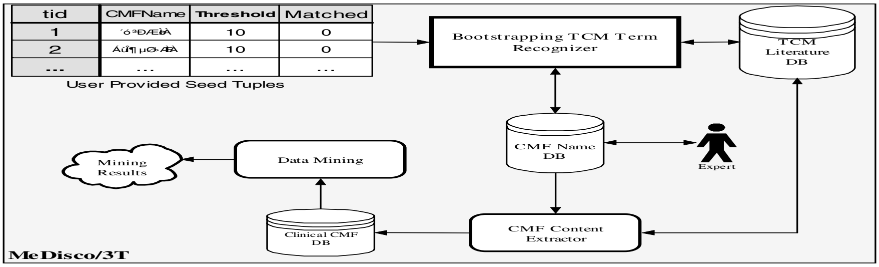
Fig 1 depicts the framework of MeDisco/3T. There are three main steps to be processed in MeDisco/3T.

[1] http://www.cintcm.com

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 396 – 398, 2005.

(1) Iterative extracts the CMF names from literature when provided with a handful of CMF seed name tuples.
(2) Extracts the CHM components and efficacy descriptions data according to the extracted CMF names. Some simple heuristic rules are used in this procedure since the abstracts are semi-structured, because most of them are delimited by special word labels such as “Approaches”, “Objectives” and “Results” etc..
(3) Conducts various kinds of data mining algorithms based on the clinical CMF database, currently, we only perform the simple frequent itemset analysis.

It is clearly that MeDisco/3T will produce two important results, namely a database of novel clinical CMFs and support of classical data mining studies on CHM.



tid	CMFName	Threshold	Matched
1	´ó³ÐÆøÌÀ	10	0
2	ÁùÎ¶µØ»ÆÌÀ	10	0
...	...	...	...

Fig. 1. MeDisco/3T text mining system

3 Main Results

We extract and identify 18,213 CMFs (from the year of 2000 to 2003 of TCM literature database) with different CHM composition to have a frequent itemset CHM analysis. The average name extraction precision by bootstrapping method[2] is high over 95%. MeDisco/3T performs a preprocessing procedure to transform the extracted data to a completely structured form, which is suitable for data mining algorithms. Where after, it apply the Apriori algorithm to analyze the frequent CHM pairs and CHM family combination characteristics in clinical CMF using.

All the clinical CMF used in TCM can be classified by its efficacy. Exactly, One CHM can be used for different efficacy in different CMF. We have chosen five different types of CMFs according to the efficacy such as HuoXueHuaYu 活血化瘀, BuZhongYiQi 补中益气 etc.. The 10 frequent CHM pairs and family combinations of the above two CMF types are listed in Table 1. Due to the page limit, the results of the other three CMF types are not depicted. It indicated from the experiment results that there exist many important CHM pairs and family combinations with different efficacy. For example, 黄芪 - 当归 is a typical CHM pair with BuZhongYiQi efficacy, and 伞形科 - 豆科 builds the core of CMF with BuZhongYiQi efficacy, because the

[2] Zhou, X., Text Mining and the Applications in TCM. PhD thesis, College of computer science, Zhejiang University, 2004,12,8. The thesis has a detail description of bootstrapping method used in MeDisco/3T.

support of 伞形科 - 豆科 combination in CMF with BuZhongYiQi efficacy is 180%. This knowledge will surely help to clinical CMF prescription practice and new drug development.

Table 1. The 10 top frequent CHM and its family combinations of efficacy BuzhongYiQi and HuoXueHuaYu, Supp(Family/CHM) represents the support of frequent family/CHM combination. For convenience, the CHM name is in Chinese, but all can be referred from the online databases on http://www.cintcm.com for the Latin or English names.

	BuZhongYiQi			HongXueHuaYu	
Family	Supp(Family/CHM)	CHM	Family	Supp(Family/CHM)	CHM
伞形科 - 豆科	1.8/0.3	当归 - 黄芪	伞形科 - 菊科	1.2/0.36	川穹 - 当归
桔梗科 - 豆科	1.1/0.26	甘草 - 党参	伞形科 - 唇形科	1.1/0.31	红花 - 桃仁
毛茛科 - 豆科	0.96/0.24	当归 - 白术	伞形科 - 豆科	1.03/0.30	桃仁 - 当归
菊科 - 豆科	0.96/0.23	党参 - 黄芪	伞形科 - 豆科	0.95/0.29	红花 - 当归
姜科 - 豆科	0.80/0.23	当归 - 柴胡	伞形科 - 蔷薇科	0.93/0.29	赤芍 - 当归
伞形科 - 菊科	0.79/0.22	白术 - 黄芪	伞形科 - 伞形科	0.77/0.28	川穹 - 桃仁
豆科 - 豆科	0.74/0.22	白术 - 党参	伞形科 - 姜科	0.69/0.26	川穹 - 红花
蔷薇科 - 豆科	0.70/0.22	甘草 - 黄芪	唇形科 - 豆科	0.67/0.26	川穹 - 赤芍
伞形科 - 桔梗科	0.63/0.21	升麻 - 黄芪	伞形科 - 芍药科	0.65/0.24	丹参 - 当归
伞形科 - 豆科	0.63/0.21	升麻 - 当归	唇形科 - 菊科	0.59/0.24	赤芍 - 桃仁

Acknowledgements

This work is partially supported by Scientific Breakthrough Program of Beijing Municipal Science&Technology Commission under grant number H020920010130.

References

1. Blagosklonny M.V., Pardee A.B., Unearthing the gems. Nature, 2002, 416(6879). 373.
2. Jenssen T.K., Lagreid A., Komorowsk J. et al., A literature network of human genes for high-throughput analysis of gene expression. Nature Genetics, vol 28,2001.pp: 21-28.
3. Bunescu R., Ge R., Rohit J.K. et al, Learning to Extract Proteins and their Interactions from Medline Abstracts. Tom Fawcett, Nina Mishra eds.,Proc. of ICML-2003 on Machine Learning in Bioinformatics, Menlo Park :AAAI Press, 2003, pp:46-53.
4. Nahm U.Y., Mooney R.J., A Mutually Beneficial Integration of Data Mining and Information Extraction. AAAI-2000, Austin, TX, pp: 627-632, July 2000.
5. Brin, S., Extracting Patterns and Relations from the World Wide Web. WebDB Workshop at EDBT-98. 1998.

Author Index