Lecture Notes in Artificial Intelligence 2226

Subseries of Lecture Notes in Computer Science
Edited by J. G. Carbonell and J. Siekmann

Lecture Notes in Computer Science
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Tokyo

Klaus P. Jantke Ayumi Shinohara (Eds.)

Discovery Science

4th International Conference, DS 2001
Washington, DC, USA, November 25-28, 2001
Proceedings

Springer

Series Editors

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

Volume Editors

Klaus P. Jantke
DFKI GmbH Saarbrücken
66123 Saarbrücken, Germany
E-mail: jantke@dfki.de

Ayumi Shinohara
Kyushu University, Department of Informatics
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan
E-mail: ayumi@i.kyushu-u.ac.jp

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Discovery science : 4th international conference ; proceedings / DS 2001, Washington, DC, USA, November 25 - 28, 2001. Klaus P. Jantke ; Ayumi Shinohara (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2226 : Lecture notes in artificial intelligence)
ISBN 3-540-42956-5

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

ISBN 3-540-42956-5 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author
Printed on acid-free paper SPIN: 10840973 06/3142 5 4 3 2 1 0

Preface

These are the conference proceedings of the 4th International Conference on Discovery Science (DS 2001). Although discovery is naturally ubiquitous in science, and scientific discovery itself has been subject to scientific investigation for centuries, the term Discovery Science is comparably new. It came up in connection with the Japanese Discovery Science project (cf. Arikawa's invited lecture on *The Discovery Science Project in Japan* in the present volume) some time during the last few years.

Setsuo Arikawa is the father in spirit of the Discovery Science conference series. He led the above mentioned project, and he is currently serving as the chairman of the international steering committee for the Discovery Science conference series. The other members of this board are currently (in alphabetical order) Klaus P. Jantke, Masahiko Sato, Ayumi Shinohara, Carl H. Smith, and Thomas Zeugmann.

Colleagues and friends from all over the world took the opportunity of meeting for this conference to celebrate Arikawa's 60th birthday and to pay tribute to his manifold contributions to science, in general, and to Learning Theory and Discovery Science, in particular.

Algorithmic Learning Theory (ALT, for short) is another conference series initiated by Setsuo Arikawa in Japan in 1990. In 1994, it amalgamated with the conference series on Analogical and Inductive Inference (AII), when ALT was held outside of Japan for the first time.

This year, ALT 2001 and DS 2001 were co-located in Washington D.C., held in parallel and sharing five invited talks and all social events. The proceedings of ALT 2001 are published as a twin volume of the present one as LNAI 2225.

The present volume is organized in three parts. The first part contains the five invited lectures of ALT 2001 and DS 2001 exactly in the order in which they appeared in the conferences' common advance program. The invited speakers are Setsuo Arikawa, Lindley Darden, Dana Angluin, Ben Shneiderman, and Paul R. Cohen. Because their talks were invited to both conferences, there had to be found a modus vivendi for publication. This volume contains the full versions of Lindley Darden's and Ben Shneiderman's paper as well as abstracts of the others.

The second part contains the accepted 30 regular papers of the DS 2001 conference. Last but not least, there is a third part with written versions of the posters accepted for presentation during the conference. In a sense, DS 2001 posters are posters of ALT 2001 as well, because both events shared a conference venue including the exhibition area for the posters.

The combination of ALT 2001 and DS 2001 allowed for an especially comprehensive treatment of the issues ranging from rather theoretical investigations to applications and to both psychological and sociological topics. The organizers consider this an attractive approach to both communities.

Over the past dozen or so years, many enterprises have begun to routinely capture paramount volumes of data describing their operations, products, services, and customers. Simultaneously, scientists and engineers have been record-

ing experimental data of a continuously growing size covering experience in many fields. The finer the measurement granularity of the engineers' equipment and the more computer power available to support scientific experiments, the larger the amounts of data captured. These huge collections of bits and bytes constitute a new challenge to those who try to separate the wheat from the chaff.

Potentially, there is more fruitful knowledge hiding in huge amounts of data, but combinatorially, there is even more rubbish. It requires a new dimension of technological investment to extract useful information, and humans must attack these problems differently. Discovery Science deals with all aspects of promoting scientific discovery, and it is changing its character within a changing world.

New questions are being asked and leading to innovative concepts. Conceptualizations are setting the stage for asking new questions. Under these circumstances, new ways of looking at the problems might arise. More traditional disciplines are invoked and innovative ideas are made precise to get computers involved in knowledge discovery. Autonomously working machinery is necessary to deal with the flood of data, thus learning becomes a core technology of discovery science. When all said and done, humans and machines must learn together and support each other.

The field of Discovery Science is evolving and frequently changing its appearance. The Discovery Science conference series aims at reflecting this development, summarizing the state of affair and helping humans to navigate in such an exciting environment.

September 2001

Klaus P. Jantke
Ayumi Shinohara

Organization

Conference Chair

Masahiko Sato	Kyoto Univ., Japan

Program Committee

Klaus P. Jantke (Co-chair)	DFKI GmbH, Germany
Ayumi Shinohara (Co-chair)	Kyushu Univ. Japan
Diane J. Cook	Univ. of Texas at Arlington, USA
Andreas Dengel	DFKI GmbH, Germany
Michael E. Gorman	Univ. of Virginia, USA
Achim Hoffmann	UNSW, Australia
John R. Josephson	Ohio State Univ., USA
Pat Langley	ISLE, USA
Hiroshi Motoda	Osaka Univ., Japan
Ryohei Nakano	Nagoya Inst. Techn., Japan
Yukio Ohsawa	Tsukuba Univ., Japan
Jorge C.G. Ramirez	Intelligent Tech. Corp., USA
Kazumi Saito	NTT, Japan
Einoshin Suzuki	Yokohama National Univ., Japan
Stefan Wrobel	Techn. Univ. Magdeburg, Germany
Thomas Zeugmann	Med. Univ. Lübeck, Germany

Local Arrangements

Carl Smith	Univ. of Maryland, USA

Subreferees

Hiroki Arimura	Maciej Liskiewicz
Markus Bläser	Shigeo Matsubara
Eisaku Maeda	Bodo Siebert
Tsutomu Hirao	Marin Simina
Armin Hust	Hirotoshi Taira
Naresh Iyer	Paul Thagard
Andreas Jakoby	Naonori Ueda
Markus Junker	Shravan Vasishth
Stefan Klink	

Table of Contents

Invited Papers

Regular Papers

Poster Papers

The Discovery Science Project in Japan

Setsuo Arikawa

Department of Informatics, Kyushu University
Fukuoka 812-8581, Japan
arikawa@i.kyushu-u.ac.jp

Abstract. The Discovery Science project in Japan in which more than sixty scientists participated was a three-year project sponsored by Grant-in-Aid for Scientific Research on Priority Area from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan. This project mainly aimed to (1) develop new methods for knowledge discovery, (2) install network environments for knowledge discovery, and (3) establish Discovery Science as a new area of Computer Science / Artificial Intelligence Study.

In order to attain these aims we set up five groups for studying the following research areas:

(A) Logic for/of Knowledge Discovery
(B) Knowledge Discovery by Inference/Reasoning
(C) Knowledge Discovery Based on Computational Learning Theory
(D) Knowledge Discovery in Huge Database and Data Mining
(E) Knowledge Discovery in Network Environments

These research areas and related topics can be regarded as a preliminary definition of Discovery Science by enumeration. Thus Discovery Science ranges over philosophy, logic, reasoning, computational learning and system developments.

In addition to these five research groups we organized a steering group for planning, adjustment and evaluation of the project. The steering group, chaired by the principal investigator of the project, consists of leaders of the five research groups and their subgroups as well as advisors from the outside of the project. We invited three scientists to consider the Discovery Science overlooking the above five research areas from viewpoints of knowledge science, natural language processing, and image processing, respectively.

The group A studied discovery from a very broad perspective, taking into account of historical and social aspects of discovery, and computational and logical aspects of discovery. The group B focused on the role of inference/reasoning in knowledge discovery, and obtained many results on both theory and practice on statistical abduction, inductive logic programming and inductive inference. The group C aimed to propose and develop computational models and methodologies for knowledge discovery mainly based on computational learning theory. This group obtained some deep theoretical results on boosting of learning algorithms and the minimax strategy for Gaussian density estimation, and also methodologies specialized to concrete problems such as algorithm for finding best subsequence patterns, biological sequence compression algorithm, text categorization, and MDL-based compression. The group D aimed to create computational strategy for speeding up the discovery process in total. For this purpose,

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 1–2, 2001.

the group D was organized with researchers working in scientific domains and researchers from computer science so that real issues in the discovery process can be exposed out and practical computational techniques can be devised and tested for solving these real issues. This group handled many kinds of data: data from national projects such as genomic data and satellite observations, data generated from laboratory experiments, data collected from personal interests such as literature and medical records, data collected in business and marketing areas, and data for proving the efficiency of algorithms such as UCI repository. So many theoretical and practical results were obtained on such a variety of data. The group E aimed to develop a unified media system for knowledge discovery and network agents for knowledge discovery. This group obtained practical results on a new virtual materialization of DB records and scientific computations that help scientists to make a scientific discovery, a convenient visualization interface that treats web data, and an efficient algorithm that extracts important information from semi-structured data in the web space.

This lecture describes an outline of our project and the main results as well as how the project was prepared. We have published and are publishing special issues on our project from several journals [5],[6],[7],[8],[9],[10]. As an activity of the project we organized and sponsored Discovery Science Conference for three years where many papers were presented by our members [2],[3],[4]. We also published annual progress reports [1], which were distributed at the DS conferences. We are publishing the final technical report as an LNAI[11].

References

1. S. Arikawa, M. Sato, T. Sato, A. Maruoka, S. Miyano, and Y. Kanada. Discovery Science Progress Report No.1 (1998), No.2 (1999), No.3 (2000). *Department of Informatics, Kyushu University.*
2. S. Arikawa and H. Motoda. Discovery Science. *LNAI, Springer* 1532, 1998.
3. S. Arikawa and K. Furukawa. Discovery Science. *LNAI, Springer* 1721, 1999.
4. S. Arikawa and S. Morishita. Discovery Science. *LNAI, Springer* 1967, 2000.
5. H. Motoda and S. Arikawa (Eds.) Special Feature on Discovery Science. *New Generation Computing,* 18(1): 13–86, 2000.
6. S. Miyano (Ed.) Special Issue on Surveys on Discovery Science. *IEICE Transactions on Information and Systems,* E83-D(1): 1–70, 2000.
7. H. Motoda (Ed.) Special Issue on Discovery Science. *Journal of Japanese Society for Artificial Intelligence,* 15(4):592–702, 2000.
8. S. Morishita and S. Miyano(Eds.) Discovery Science and Data Mining (in Japanese). *bit special volume , Kyoritsu Shuppan,* 2000.
9. S. Arikawa, M. Sato, T. Sato, A. Maruoka, S. Miyano, and Y. Kanada. The Discovery Science Project. *Journal of Japanese Society for Artificial Intelligence,* 15(4) 595–607, 2000.
10. S. Arikawa, H. Motoda, K. Furukawa, and S. Morishita (Eds.) Theoretical Aspects of Discovery Science. *Theoretical Computer Science* (to appear)
11. S. Arikawa and A. Shinohara (Eds.) Progresses in Discovery Science. *LNAI, Springer* (2001, to appear)

Discovering Mechanisms: A Computational Philosophy of Science Perspective

Lindley Darden

Department of Philosophy
University of Maryland
College Park, MD 20742
darden@carnap.umd.edu
http://www.inform.umd.edu/PHIL/faculty/LDarden/

Abstract. A task in the philosophy of discovery is to find reasoning strategies for discovery, which fall into three categories: strategies for generation, evaluation and revision. Because mechanisms are often what is discovered in biology, a new characterization of mechanism aids in their discovery. A computational system for discovering mechanisms is sketched, consisting of a simulator, a library of mechanism schemas and components, and a discoverer for generating, evaluating and revising proposed mechanism schemas. Revisions go through stages from how possibly to how plausibly to how actually.

1 Introduction

Philosophers of discovery look for reasoning strategies that can guide discovery. This work is in the framework of Herbert Simon's (1997) view of discovery as problem solving. Given a problem to be solved, such as explaining a phenomenon, one goal is to find a mechanism that produces that phenomenon. For example, given the phenomenon of the production of a protein, the goal is to find the mechanism of protein synthesis. The task of the philosopher of discovery is to find reasoning strategies to guide such discoveries. Strategies are heuristics for problem solving; that is, they provide guidance but do not guarantee success.

Discovery is not viewed as something that occurs in a single a-ha moment of insight. Instead, discovery is construed as a process that occurs over an extended period of time, going through cycles of generation, evaluation, and revision (Darden 1991).

The history of science is a source of "compiled hindsight" (Darden 1987) about reasoning strategies for discovering mechanisms. This paper will use examples from the history of biology to illustrate general reasoning strategies for discovering mechanisms. Section 2 puts this work into the broader context of a matrix of biological knowledge. Section 3 discusses a new characterization of mechanism, based on an ontology of entities, properties, and activities. Section 4 outlines components of a mechanism discovery system, including a simulator, a library of mechanism designs and components, and a discoverer.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 3–15, 2001.

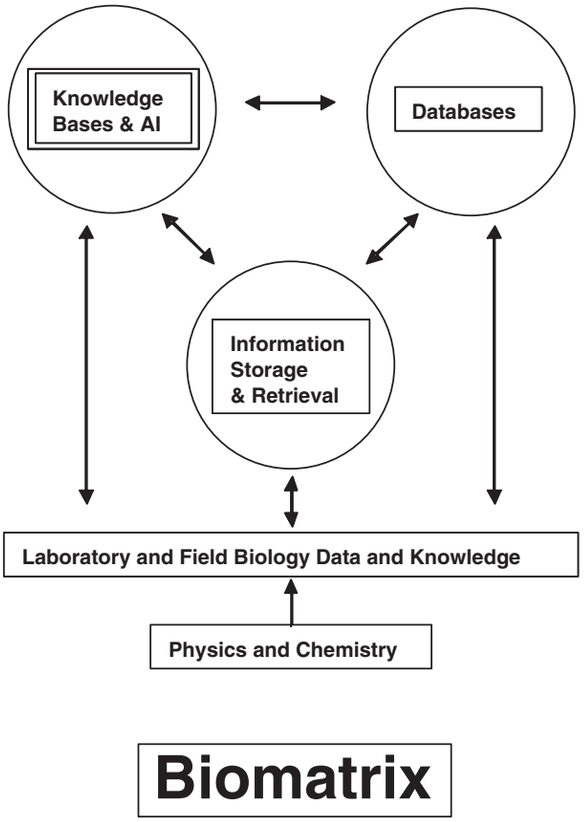


Fig. 1. Matrix of Biological Knowledge

2 Biomatrix

This work is situated in a larger framework. In the 1980s, Harold Morowitz (1985) chaired a National Academy of Sciences workshop on models in biology. As a result of that workshop, a society was formed with the name, "Biomatrix: A Society for Biological Computation and Informatics" (Morowitz and Smith 1987). This society was ahead of its time; it has splintered into different groups and its grand vision has yet to be realized. Nonetheless, its vision is worth revisiting in order to put the work to be discussed in this paper into a broader context. As Figure 1 shows, the biomatrix vision included relations among three areas: first, databases; second, information storage and retrieval by literature cataloging (e.g., Medline); and, third, artificial intelligence and knowledge bases. Discovery science has worked in all three areas since the 1980s. Knowledge discovery in databases is a booming area (e.g., Piatetsky-Shapiro and Frawley, eds., 1991).

Discovery using abstracts available from literature catalogues has been developed by Don Swanson (1990) and others. The area of discovery using knowledge based systems is an active area, especially in computational biology. The meetings on Intelligent Systems in Molecular Biology and the International Society for Computational Biology arose from that part of the biomatrix. It is in the knowledge based systems box that my work today will fall. Relations to databases and information retrieval as related to mechanism discovery will perhaps occur to the reader.

3 Mechanisms, Schemas, and Sketches

Often in biology, what is to be discovered is a mechanism. Physicists often aim to discover general laws, such as Newton's laws of motion. However, few biological phenomena are best characterized by universal, mathematical laws (Beatty 1995). The field of molecular biology, for example, studies mechanisms, such as the mechanisms of DNA replication, protein synthesis, and gene regulation. The lively area of functional genomics is now attempting to discover the mechanisms in which the gene sequences act. Such mechanisms include gene expression, during both embryological development and normal gene activities in the adult. The field of biochemistry also studies mechanisms when it find the activities that transform one stage in a pathway to next, such as the enzymes, reactants and products in the Krebs cycle that produces the energy molecule ATP. An important current scientific task is to connect genetic mechanisms studied by molecular biology with metabolic mechanisms studied by biochemistry. As that task is accomplished, science will have a unified picture of the mechanisms that carry out the two essential features of life according to Aristotle: reproduction and nutrition.

Given this importance of mechanisms in biology, a correspondingly important task for discovery science is to find methods for discovering mechanisms. If the goal is to discover a mechanism, then the nature of that product shapes the process of discovery. A new characterization of mechanism aids the search for reasoning strategies to discover mechanisms.

A mechanism is sought to explain how a *phenomenon* is produced (Machamer, Darden, Craver 2000) or how some *task* is carried out (Bechtel and Richardson 1993) or how the mechanism as a whole *behaves* (Glennan 1996). Mechanisms may be characterized in the following way:

> Mechanisms are entities and activities organized such that they are productive of regular changes from start or set-up to finish or termination conditions. (Machamer, Darden, Craver 2000, p. 3).

Mechanisms are regular in that they usually work in the same way under the same conditions. The regularity is exhibited in the typical way that the mechanism runs from beginning to end; what makes it regular is the *productive continuity* between stages. Mechanisms exhibit productive continuity without gaps

from the set up to the termination conditions; that is, each stage gives rise to, allows, drives, or makes the next.

The ontology proposed here consists of entities, properties, and activities. Mechanisms are composed of both *entities* (with their properties) and *activities*. Activities are the producers of change. Entities are the things that engage in activities. Activities require that entities have specific types of properties. For example, two entities, a DNA base and its complement, engage in the activity of hydrogen bonding because of their properties of geometric shape and weak polar charges.

For a given scientific field, there are typically entities and activities that are accepted as relatively fundamental or taken to be unproblematic for the purposes of a given scientist, research group, or field. That is, descriptions of mechanisms in that field typically bottom out somewhere. Bottoming out is relative: different types of entities and activities are where a given field stops when constructing its descriptions of mechanisms. In molecular biology, mechanisms typically bottom out in descriptions of the activities of cell organelles, such as the ribosome, and molecules, including macromolecules, smaller molecules, and ions. The most important kinds of activities in molecular biology are geometrico-mechanical and electro-chemical activities. An example of a geometrico-mechanical activity is the lock and key docking of an enzyme and its substrate. Electro-chemical activities include strong covalent bonding and weak hydrogen bonding.

Entities and activities are interdependent (Machamer, Darden, Craver 2000, p. 6). For example, appropriate chemical valences are necessary for covalent bonding. Polar charges are necessary for hydrogen bonding. Appropriate shapes are necessary for lock and key docking. This interdependence of entities and activities allows one to reason about one, based on what is known or conjectured about the other, in each stage of the mechanism (Darden and Craver, in press).

A *mechanism schema* is a truncated abstract description of a mechanism that can be filled with more specific descriptions of component entities and activities. An example is the following:

DNA → RNA → protein.

This is a diagram of the central dogma of molecular biology. It is a very abstract, schematic representation of the mechanism of protein synthesis.

A schema may be even more abstract if it merely indicates functional roles played in the mechanism by fillers of a place in the schema (Craver 2001). Consider the schema

DNA → template → protein.

The schema term "template" indicates the functional role played by the intermediate between DNA and protein. Hypotheses about role-fillers changed during the incremental discovery of the mechanism of protein synthesis in the 1950s and 1960s. Thus, mechanism schemes are particularly good ways of representing functional roles. (For discussion of "local" and "integrated" functions and a less schematic way of representing them in a computational system, see Karp 2000.)

Table 1. Constraints on the Organization of Mechanisms

Character of phenomenon

Componency Constraints
- Entities and activities
- Modules

Spatial Constraints
- Compartmentalization
- Localization
- Connectivity
- Structural
- Orientation

Temporal Constraints
- Order
- Rate
- Duration
- Frequency

Hierarchical Constraints
- Integration of levels

(from Craver and Darden 2001)

Mechanism *sketches* are incomplete schemas. They contain black boxes, which cannot yet be filled with known components. Attempts to instantiate a sketch would leave a gap in the productive continuity; that is, knowledge of the needed particular entities and activities is missing. Thus, sketches indicate what needs to be discovered in order to find a mechanism schema.

Once a schema is found and instantiated, a detailed description of a mechanism results. For example, a more detailed description of the protein synthesis mechanism (often depicted in diagrams) satisfies the constraints that any adequate description of a mechanism must satisfy. It shows how the phenomenon, the synthesis of a protein, is carried out by the operation of the mechanism. It depicts the entities–DNA, RNA, and amino acids–as well as implicitly, the activities. Hydrogen bonding is the activity operating when messenger RNA is copied from DNA. There is a geometrico-mechanical docking of the messenger RNA and the ribosome, a particle in the cytoplasm. Hydrogen bonding again occurs as the codons on messenger RNA bond to the anticodons on transfer RNAs carrying amino acids. Finally, covalent bonding is the activity that links the amino acids together in the protein. Good mechanism descriptions show the spatial relations of the components and the temporal order of the stages.

A detailed description of a mechanism satisfies several general constraints. (They are listed in Table 1 and indicated here by italics.) There is a *phenomenon* that the mechanism, when working, produces, for example, the synthesis of a pro-

tein. The nature of the phenomenon, which may be recharacterized as research on it proceeds, constrains details about the mechanism that produces it. For example, the *components* of the mechanism, the entities and activities, must be adequate to synthesize a protein, composed of amino acids tightly covalently bonded to each other. There are various *spatial constraints*. The DNA is *located* in the nucleus (in eucaryotes) and the rest of the machinery is in the cytoplasm. The ribosome is a particle with a two part *structure* that allows it to attach to the messenger RNA and *orient* the codons of the messenger so that particular transfer RNAs can hydrogen bond to them. There is a particular *order* in which the steps occur and they take certain amounts of *time*. All of these constraints can play roles in the search for mechanisms, and, then, they become part of an adequate description of a mechanism. (For more discussion of these constraints, see Craver and Darden 2001.)

From this list of constraints on an adequate description of a mechanism, it is evident that mere equations do not adequately represent the numerous features of a mechanism, especially spatial constraints. Diagrams that depict structural features, spatial relations and temporal sequences are good representations of mechanisms.

To sum up so far: Recent work has provided this new characterization of what a mechanism is, the constraints that any adequate description of a mechanism must satisfy, and an analysis of abstract mechanism schemas and incomplete mechanism sketches that can play roles in guiding discovery.

4 Outline of a System for Constructing Hypothetical Mechanisms

Components of a computational system for discovering mechanisms are outlined in Figure 2. They include a simulator, a hypothesized mechanism schema, a discoverer with reasoning strategies for generation, evaluation, and revision, and a searchable, indexed library.

4.1 Simulator

The goal is to construct a simulator that adequately simulates a biological mechanism. Given the set up conditions, the simulator can be used to predict specific termination conditions. The simulator is an instantiation of a mechanism schema. It may contain more or less detail about the specific component entities and activities and their structural, spatial and temporal organization. From a human factors perspective, a video option to display the mechanism simulation in action would aid the user in seeing what the mechanism is doing at each stage. The video could be stopped at each stage and details of the entities and activities of that stage examined in more detail.

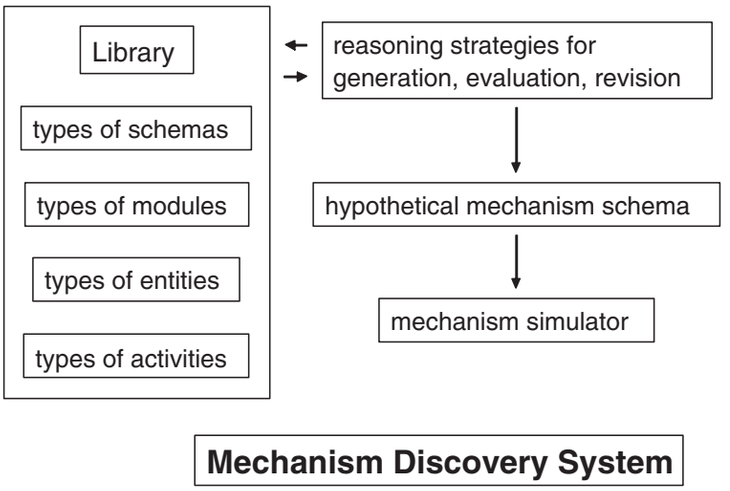


Fig. 2. Outline for a Mechanism Discovery System

4.2 Library

A mechanism schema is discovered by iterating through stages of generation, evaluation, and revision. Generation is accomplished by several steps. First, a phenomenon to be explained must be characterized. Its mode of description will guide the search for schemas that can produce it. Search occurs within a library, consisting of several types of entries: types of schemas, types of modules, types of entities, and types of activities.

The search among types of schemas is a search for an abstraction of an analogous mechanism (on analogies and schemas, see, e.g., Holyoak and Thagard 1995). Kevin Dunbar (1995) has shown that molecular biologists often use "local analogies" to similar mechanisms in their own field and "regional analogies" to mechanisms in other, neighboring fields. Such analogies are good sources from which to abstract mechanism schemas.

Types of schemas, modules, entities and activities are interconnected. A particular type of schema, for example, a gene regulation schema, may suggest one or more types of modules, such as derepression or negative feedback modules. A type of entity will have activity-enabling properties that indicate it can produce a type of activity. Conversely, a type of activity will require particular types of entities. For example, nucleic acids have polar charged bases that enable them to engage in the activity of hydrogen bonding, a weak form of chemical bonding that can be easily formed and broken between polar molecules.

Schemas may be indexed by the kind of phenomenon they produce. For example, for the phenomenon of producing an adaptation, two types of mechanisms have been proposed historically by biologists–selective mechanisms and instructive mechanisms (Darden, 1987). At a high degree of abstraction, a selection

schema may be characterized as follows: first comes a stage of variant production; next comes a stage with a selective interaction that poses a challenge to the variants; this is followed by differential benefit for some of the variants. In contrast, instructive mechanisms have a coupling between the stage of variant production and the selective environment, so that an instruction is sent from the environment and interpreted by the adaptive system to produce only the required variant. In evolutionary biology and immunology, selective mechanisms rather than instructive ones have been shown to work in producing evolutionary adaptations and clones of antibody cells (Darden and Cain 1989).

A library of modules can be indexed by the functional roles they can fulfill in a schema (e.g., Goel and Chandrasekaran 1989). For example, if a schema requires end-product inhibition, then a feedback control module can be added to the linear schema. If cell-to-cell signaling is indicated, then membrane spanning proteins serving as receptors are a likely kind of module. Entities and activities can be categorized in numerous ways. Types of macromolecules include nucleic acids, proteins, and carbohydrates. When proteins, for example, perform functions, such as enzymes that catalyze reactions, then the kind of function, such as phosphorylation, is a useful indexing method.

4.3 Discoverer: Generation, Evaluation, Revision

During generation, after a phenomenon is characterized, then a search is made to see if an entire schema can be found that produces such a type of phenomenon. If an entire schema can be found, such as a selective or instructive schema, then generation can proceed to further specification with types of modules, entities, and activities. If no entire schema is available, then modules may be put together piecemeal to fulfill various functional roles. If functional roles and modules to fill them are not yet known, then reasoning about types of entities and activities is available. By starting from known set up conditions, or, conversely, from the end product, a hypothesized string of entities and activities can be constructed. Reasoning forward from the beginning or backward from the end product of the mechanism will allow gaps in the middle to be filled. In sum, reasoning strategies for discovering mechanisms include schema instantiation, modular subassembly, and forward chaining/backtracking (Darden, forthcoming).

Evaluation. Once one or more hypothesized mechanism schemas are found or constructed piecemeal, then evaluation occurs. Evaluation proceeds through stages from how possibly to how plausibly to how actually. (Peter Machamer suggests that "how actually" is best read as "how most plausibly," given that all scientific claims are contingent, that is, subject to revision in the light of new evidence.)

How possibly a mechanism operates can be shown by building a simulator that begins with the set up conditions and produces the termination conditions by moving through hypothesized intermediate stages. As additional constraints are fulfilled and evaluation strategies applied, the proposed mechanism becomes

Table 2. Strategies for Theory Evaluation

1. Internally consistent and nontautologous
2. Systematicity vs. modularity
3. Clarity
4. Explanatory adequacy
5. Predictive adequacy
6. Scope and generality
7. Lack of ad hocness
8. Extendability and fruitfulness
9. Relations with other accepted theories
10. Metaphysical and methodological constraints
11. Relation to rivals

(from Darden 1991, p. 257)

more plausible. The constraints of Table 1 must be satisfied. Table 2 (from Darden 1991, Table 15-2) lists strategies for theory assessment often employed by philosophers of science. A working simulator will likely show that the proposed schema is internally consistent and consists of modules whose functioning is clearly understood, thus satisfying some of the conditions listed in 1-3. If the simulator can be run to produce the phenomenon to be explained, then condition 4 of explanatory adequacy is at least partially fulfilled. Testing a prediction against data is often viewed as the most important evaluation strategy. The simulator can be run with different initial conditions to produce predictions, which can be tested against data. If a prediction does not match a data point, then an anomaly results and revision is required. We will omit further discussion of the other strategies for theory assessment in order to turn our attention to anomaly resolution strategies to use when revision is required.

Anomaly resolution. When a prediction does not match a data point, then an anomaly results. Strategies for anomaly resolution require a number of information processing tasks to be carried out. In previous work with John Josephson and Dale Moberg, we investigated computational implementation of such tasks (Moberg and Josephson 1990; Darden et al. 1992; Darden 1998). A list of such tasks is found in Figure 3.

Reasoning in anomaly resolution is, first, a diagnostic reasoning task, to localize the site(s) of failure, and, then, a redesign task, to improve the simulation to remove the problem. Characterizing the exact difference between the prediction and the data point is a first step. Peter Karp (1990; 1993) discussed this step of anomaly resolution in his implementation of the MOLGEN system to resolve anomalies in a molecular biology simulator. One wants to milk the anomaly itself for all the information one can get about the nature of the failure. Often during diagnosis, the nature of the anomaly allows failures to be localized to one part of the system rather than others, sometimes to a specific site.

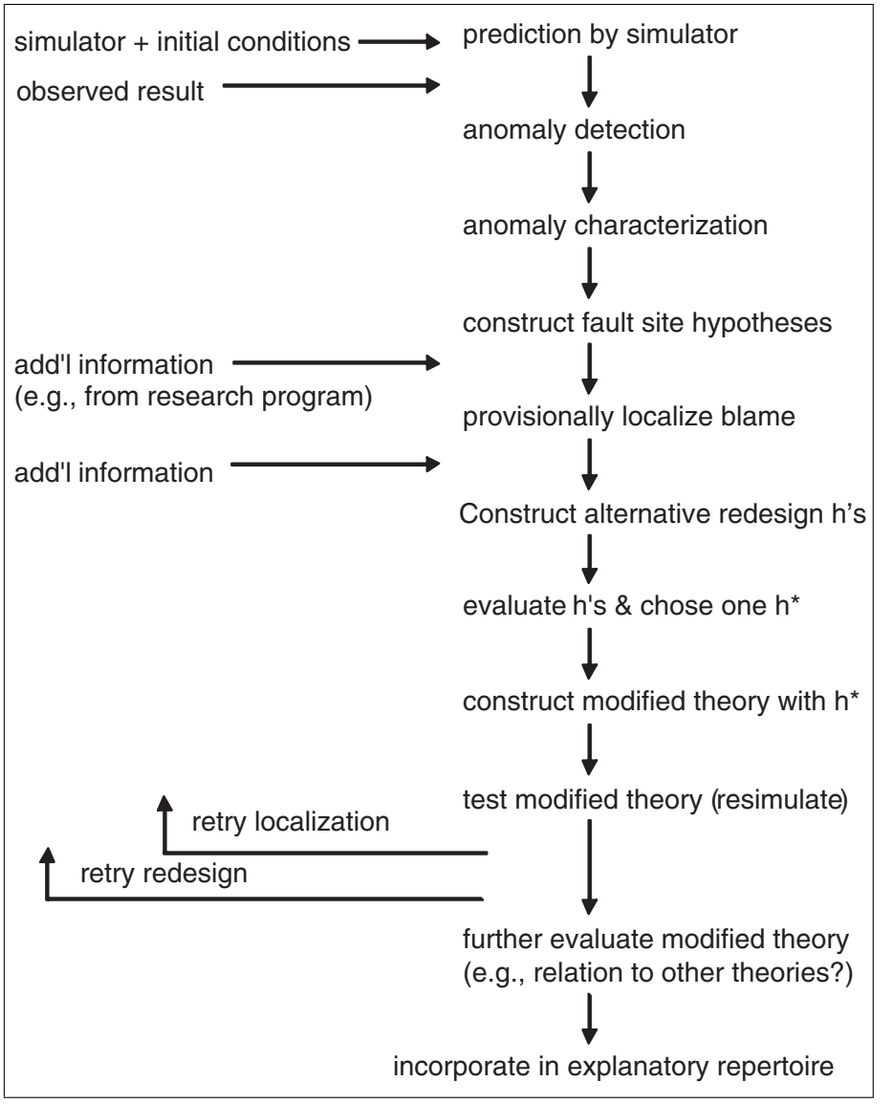


Fig. 3. Information Processing Tasks in Anomaly Resolution (from Darden 1998, p. 69)

Once hypothesized localizations are found by doing credit assignment, then alternative redesign hypotheses for that module can be constructed. Once again, the library of modules, entities and activities can be consulted to find plausible candidates. The newly redesigned simulator can be run again to see if the problem is fixed and the prediction now matches the data point.

5 Piecemeal Discovery and Hierarchical Integration

The view of scientific discovery proposed here is that discovery of mechanisms occurs in extended episodes of cycles of generation, evaluation, and revision. In so far as the constraints are satisfied, the assessment strategies are applied, and any anomalies are resolved, then the hypothesized mechanism will have moved through the stages of how possibly to how plausibly to how actually. A new mechanism will have been discovered.

Once a new mechanism at a given mechanism level has been discovered, then that mechanism needs to be situated within the context of other biological mechanisms. Thus, the general strategy for theory evaluation of consistent relations with other accepted theories in other fields of science (see Table 2, strategy 9) is reinterpreted. By thinking about theories as mechanism schemas, the strategy gets implemented by situating the hypothesized mechanism in a larger context. This larger context consists of mechanisms that occur before and after it, as well as mechanisms up or down in a mechanism hierarchy (Craver 2001). Biological mechanisms are nested within other mechanisms, and finding such a fit in an integrated picture is another measure of the adequacy of a newly proposed mechanism.

6 Conclusion

Integrated mechanism schemas can serve as the scaffolding of the biological matrix. They provide a framework to integrate general biological knowledge of mechanisms, the data that provide evidence for such mechanisms, and the reports in the literature of research to discover mechanisms.

This paper has discussed a new characterization of mechanism, based on an ontology of entities, properties, and activities, and has outlined components of a computational system for discovering mechanisms. Discovery is viewed as an extended process, requiring reasoning strategies for generation, evaluation, and revision of hypothesized mechanism schemas. Discovery moves through the stages of from how possibly to how plausibly to how actually a mechanism works.

Acknowledgments

This work was supported by the National Science Foundation under grant number SBR-9817942. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect

those of the National Science Foundation. Many of the ideas in this paper were worked out in collaboration with Carl Craver and Peter Machamer.

References

1. Beatty, John (1995), "The Evolutionary Contingency Thesis," in James G. Lennox and Gereon Wolters (eds.), *Concepts, Theories, and Rationality in the Biological Sciences*. Pittsburgh, PA: University of Pittsburgh Press, pp. 45-81.
2. Bechtel, William and Robert C. Richardson (1993), Discovering Complexity: *Decomposition and Localization as Strategies in Scientific Research*. Princeton, N. J.: Princeton University Press.
3. Craver, Carl (2001), "Role Functions, Mechanisms, and Hierarchy," *Philosophy of Science* 68: 53-74.
4. Craver, Carl and Lindley Darden (2001), "Discovering Mechanisms in Neurobiology: The Case of Spatial Memory," in Peter Machamer, R. Grush, and P. McLaughlin (eds.), *Theory and Method in the Neurosciences*. Pittsburgh, PA: University of Pittsburgh Press, pp. 112-137.
5. Darden, Lindley (1987), "Viewing the History of Science as Compiled Hindsight," *AI Magazine* 8(2): 33-41.
6. Darden, Lindley (1990), "Diagnosing and Fixing Faults in Theories," in J. Shrager and P. Langley (eds.), *Computational Models of Scientific Discovery and Theory Formation*. San Mateo, CA: Morgan Kaufmann, pp. 319-346.
7. Darden, Lindley (1991), *Theory Change in Science: Strategies from Mendelian Genetics*. New York: Oxford University Press.
8. Darden, Lindley (1998), "Anomaly-Driven Theory Redesign: Computational Philosophy of Science Experiments," in Terrell W. Bynum and James Moor (eds.), *The Digital Phoenix: How Computers are Changing Philosophy*. Oxford: Blackwell, pp. 62-78. Available: www.inform.umd.edu/PHIL/faculty/LDarden/Research/pubs/
9. Darden, Lindley (forthcoming), "Strategies for Discovering Mechanisms: Schema Instantiation, Modular Subassembly, Forward Chaining/Backtracking," Presented at PSA 2000, Vancouver. Preprint available: www.inform.umd.edu/PHIL/faculty/LDarden/Research/pubs
10. Darden, Lindley and Joseph A. Cain (1989), "Selection Type Theories," Philosophy of Science 56: 106-129.
    Available: www.inform.umd.edu/PHIL/faculty/LDarden/Research/pubs/
11. Darden, Lindley and Carl Craver (in press), "Strategies in the Interfield Discovery of the Mechanism of Protein Synthesis," *Studies in History and Philosophy of Biological and Biomedical Sciences*.
12. Darden, Lindley, Dale Moberg, Sunil Thadani, and John Josephson, (July 1992), "A Computational Approach to Scientific Theory Revision: The TRANSGENE Experiments," Technical Report 92-LD-TRANSGENE, Laboratory for Artificial Intelligence Research, The Ohio State University. Columbus, Ohio, USA.
13. Dunbar, Kevin (1995), "How Scientists Really Reason: Scientific Reasoning in Real-World Laboratories," in R. J. Sternberg and J. E. Davidson (eds.), *The Nature of Insight*. Cambridge, MA: MIT Press, pp. 365-395.
14. Glennan, Stuart S. (1996), "Mechanisms and The Nature of Causation," *Erkenntnis* 44: 49-71.
15. Goel, Ashok and B. Chandrasekaran, (1989) "Functional Representation of Designs and Redesign Problem Solving," in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI, August 1989, pp. 1388-1394.

16. Holyoak, Keith J. and Paul Thagard (1995), *Mental Leaps: Analogy in Creative Thought*. Cambridge, MA: MIT Press.
17. Karp, Peter (1990), "Hypothesis Formation as Design," in J. Shrager and P. Langley (eds.), *Computational Models of Scientific Discovery and Theory Formation*. San Mateo, CA: Morgan Kaufmann, pp. 275-317.
18. Karp, Peter (1993), "A Qualitative Biochemistry and its Application to the Regulation of the Tryptophan Operon," in L. Hunter (ed.), *Artificial Intelligence and Molecular Biology*. Cambridge, MA: AAAI Press and MIT Press, pp. 289-324.
19. Karp, Peter D. (2000), "An Ontology for Biological Function Based on Molecular Interactions," *Bioinformatics* 16:269-285.
20. Machamer, Peter, Lindley Darden, and Carl Carver (2000), "Thinking About Mechanisms," *Philosophy of Science* 67: 1-25.
21. Moberg, Dale and John Josephson (1990), "Diagnosing and Fixing Faults in Theories, Appendix A: An Implementation Note," in J. Shrager and P. Langley (eds.), *Computational Models of Scientific Discovery and Theory Formation*. San Mateo, CA: Morgan Kaufmann, pp. 347-353.
22. Morowitz, Harold (1985), "Models for Biomedical Research: A New Perspective," Report of the Committee on Models for Biomedical Research. Washington, D.C.: National Academy Press.
23. Morowitz, Harold and Temple Smith (1987), "Report of the Matrix of Biological Knowledge Workshop, July 13-August 14, 1987," Sante Fe, NM: Sante Fe Institute.
24. Piatetsky-Shapiro, Gregory and William J. Frawley (eds.) (1991), *Knowledge Discovery in Databases*. Cambridge, MA: MIT Press.
25. Simon, Herbert A. (1977), *Models of Discovery*. Dordrecht: Reidel.
26. Swanson, Don R. (1990), "Medical Literature as a Potential Source of New Knowledge," *Bull. Med. Libr. Assoc.* 78:29-37.

Queries Revisited

Dana Angluin

Computer Science Department
Yale University
P. O. Box 208285
New Haven, CT 06520-8285
angluin@cs.yale.edu

Abstract. We begin with a brief tutorial on the problem of learning a finite concept class over a finite domain using membership queries and/or equivalence queries. We then sketch general results on the number of queries needed to learn a class of concepts, focusing on the various notions of combinatorial dimension that have been employed, including the teaching dimension, the exclusion dimension, the extended teaching dimension, the fingerprint dimension, the sample exclusion dimension, the Vapnik-Chervonenkis dimension, the abstract identification dimension, and the general dimension.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, p. 16, 2001.

Inventing Discovery Tools: Combining Information Visualization with Data Mining

Ben Shneiderman

Department of Computer Science, Human-Computer Interaction Laboratory, Institute for Advanced Computer Studies, and Institute for Systems Research University of Maryland, College Park, MD 20742 USA ben@cs.umd.edu*

Abstract. The growing use of information visualization tools and data mining algorithms stems from two separate lines of research. Information visualization researchers believe in the importance of giving users an overview and insight into the data distributions, while data mining researchers believe that statistical algorithms and machine learning can be relied on to find the interesting patterns. This paper discusses two issues that influence design of discovery tools: statistical algorithms vs. visual data presentation, and hypothesis testing vs. exploratory data analysis. I claim that a combined approach could lead to novel discovery tools that preserve user control, enable more effective exploration, and promote responsibility.

1 Introduction

Genomics researchers, financial analysts, and social scientists hunt for patterns in vast data warehouses using increasingly powerful software tools. These tools are based on emerging concepts such as knowledge discovery, data mining, and information visualization. They also employ specialized methods such as neural networks, decisions trees, principal components analysis, and a hundred others.

Computers have made it possible to conduct complex statistical analyses that would have been prohibitive to carry out in the past. However, the dangers of using complex computer software grow when user comprehension and control are diminished. Therefore, it seems useful to reflect on the underlying philosophy and appropriateness of the diverse methods that have been proposed. This could lead to better understandings of when to use given tools and methods, as well as contribute to the invention of new discovery tools and refinement of existing ones.

Each tool conveys an outlook about the importance of human initiative and control as contrasted with machine intelligence and power [16]. The conclusion deals with the central issue of responsibility for failures and successes. Many issues influence design of discovery tools, but I focus on two: statistical algorithms vs. visual data presentation and hypothesis testing vs. exploratory data analysis.

* Keynote for Discovery Science 2001 Conference, November 25-28, 2001, Washington, DC. Also to appear in Information Visualization, new journal by Palgrave/MacMillan.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 17–28, 2001.

2 Statistical Algorithms vs. Visual Data Presentation

Early efforts to summarize data generated means, medians, standard deviations, and ranges. These numbers were helpful because their compactness, relative to the full data set, and their clarity supported understanding, comparisons, and decision making. Summary statistics appealed to the rational thinkers who were attracted to the objective nature of data comparisons that avoided human subjectivity. However, they also hid interesting features such as whether distributions were uniform, normal, skewed, bi-modal, or distorted by outliers. A remedy to these problems was the presentation of data as a visual plot so interesting features could be seen by a human researcher.

The invention of times-series plots and statistical graphics for economic data is usually attributed to William Playfair (1759-1823) who published The Commercial and Political Atlas in 1786 in London. Visual presentations can be very powerful in revealing trends, highlighting outliers, showing clusters, and exposing gaps. Visual presentations can give users a richer sense of what is happening in the data and suggest possible directions for further study. Visual presentations speak to the intuitive side and the sense-making spirit that is part of exploration. Of course visual presentations have their limitations in terms of dealing with large data sets, occlusion of data, disorientation, and misinterpretation.

By early in the 20th century statistical approaches, encouraged by the Age of Rationalism, became prevalent in many scientific domains. Ronald Fisher (1890-1962) developed modern statistical methods for experimental designs related to his extensive agricultural studies. His development of analysis of variance for design of factorial experiments [7] helped advance scientific research in many fields [12]. His approaches are still widely used in cognitive psychology and have influenced most experimental sciences.

The appearance of computers heightened the importance of this issue. Computers can be used to carry out far more complex statistical algorithms and they also be used to generate rich visual, animated, and user-controlled displays. Typical presentation of statistical data mining results is by brief summary tables, induced rules, or decision trees. Typical visual data presentations show data-rich histograms, scattergrams, heatmaps, treemaps, dendrograms, parallel coordinates, etc. in multiple coordinated windows that support user-controlled exploration with dynamic queries for filtering (Fig. 1). Comparative studies of statistical summaries and visual presentations demonstrate the importance of user familiarity and training with each approach and the influence of specific tasks. Of course, statistical summaries and visual presentations can both be misleading or confusing.

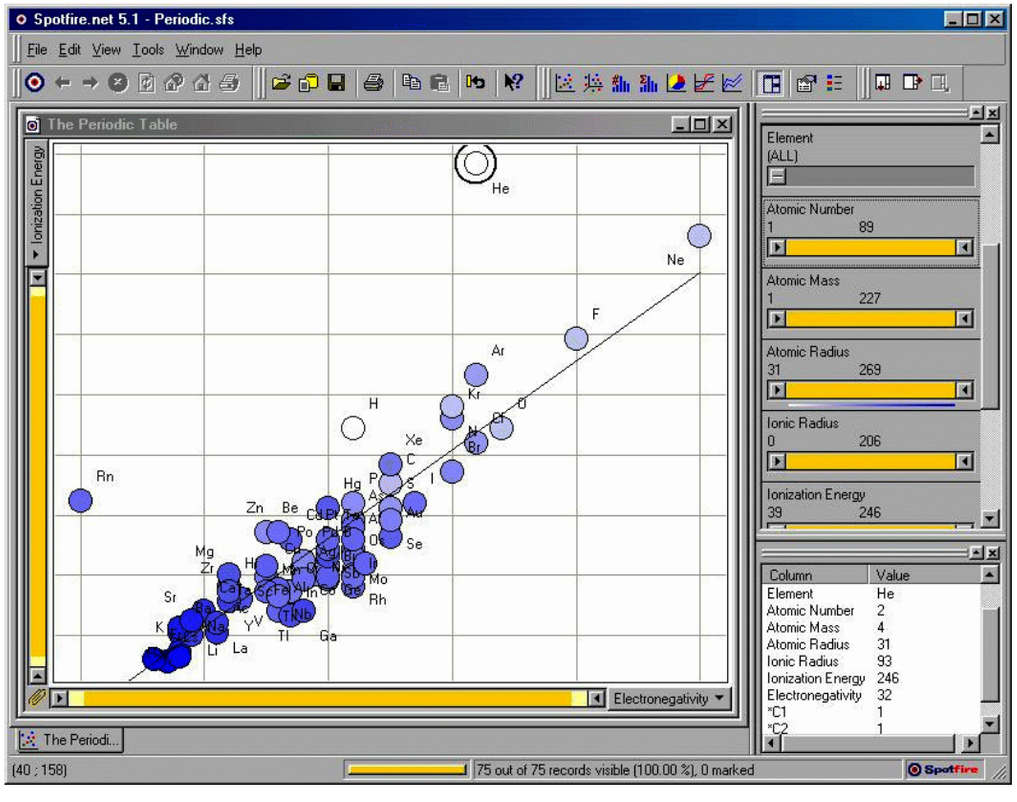


Fig. 1. Spotfire (www.spotfire.com) display of chemical elements showing the strong correlation between ionization energy and electronegativity, and two dramatic outliers: radon and helium.

An example may help clarify the distinction. Promoters of statistical methods may use linear correlation coefficients to detect relationships between variables, which works wonderfully when there is a linear relationship between variables and when the data is free from anomalies. However, if the relationship is quadratic (or exponential, sinusoidal, etc.) a linear algorithm may fail to detect the relationship. Similarly if there are data collection problems that add outliers or if there are discontinuities over the range (e.g. freezing or boiling points of water), then linear correlation may fail. A visual presentation is more likely to help researchers find such phenomena and suggest richer hypotheses.

3 Hypothesis Testing vs. Exploratory Data Analysis

Fisher's approach not only promoted statistical methods over visual presentations, but also strongly endorsed theory-driven hypothesis-testing research over casual observation and exploratory data analysis. This philosophical strand goes back to Francis Bacon (1551-1626) and later to John Herschel's 1830 A Preliminary Discourse on the Study of Natural Philosophy. They are usually credited with influencing modern notions of scientific methods based on rules of induction and the hypothetico-deductive method. Believers in scientific methods typically see controlled experiments as the fast path to progress, even though its use of the reductionist approach to test one variable at a time can be disconcertingly slow. Fisher's invention of factorial experiments helped make controlled experimentation more efficient.

Advocates of the reductionist approach and controlled experimentation argue that large benefits come when researchers are forced to clearly state their hypotheses in advance of data collection. This enables them to limit the number of independent variables and to measure a small number of dependent variables. They believe that the courageous act of stating hypotheses in advance sharpens thinking, leads to more parsimonious data collection, and encourages precise measurement. Their goals are to understand causal relationships, to produce replicable results, and to emerge with generalizable insights. Critics complain that the reductionist approach, with its laboratory conditions to ensure control, is too far removed from reality (not situated and therefore stripped of context) and therefore may ignore important variables that effect outcomes. They also argue that by forcing researchers to state an initial hypothesis, their observation will be biased towards finding evidence to support their hypothesis and will ignore interesting phenomena that are not related to their dependent variables.

On the other side of this interesting debate are advocates of exploratory data analysis who believe that great gains can be made by collecting voluminous data sets and then searching for interesting patterns. They contend that statistical analyses and machine learning techniques have matured enough to reveal complex relationships that were not anticipated by researchers. They believe that a priori hypotheses limit research and are no longer needed because of the capacity of computers to collect and analyze voluminous data. Skeptics worry that any given set of data, no matter how large, may still be a special case, thereby undermining the generalizability of the results. They also question whether detection of strong statistical relationships can ever lead to an understanding of cause and effect. They declare that correlation does not imply causation.

Once again, an example may clarify this issue. If a semiconductor fabrication facility is generating a high rate of failures, promoters of hypothesis testing might list the possible causes, such as contaminants, excessive heat, or too rapid cooling. They might seek evidence to support these hypotheses and maybe conduct trial runs with the equipment to see if they could regenerate the problem. Promoters of exploratory data analysis might want to collect existing data from the past year of production under differeing conditions and then run data mining tools against these data sets to discover correlates of high rates of failure. Of course, an experienced supervisor may blend these approaches, gathering exploratory hypotheses from the existing data and then conducting confirmatory tests.

4 The New Paradigms

The emergence of the computer has shaken the methodological edifice. Complex statistical calculations and animated visualizations become feasible. Elaborate controlled experiments can be run hundreds of times and exploratory data analysis has become widespread. Devotees of hypothesis-testing have new tools to collect data and prove their hypotheses. T-tests and analysis of variance (ANOVA) have been joined by linear and non-linear regression, complex forecasting methods, and discriminant analylsis.

Those who believe in exploratory data analysis methods have even more new tools such as neural networks, rule induction, a hundred forms of automated clustering, and even more machine learning methods. These are often covered in the rapidly growing academic discipline of data mining [6,8]. Witten and Frank define data mining as "the extraction of implicit, previously unknown, and potentially useful information from data." They caution that "exaggerated reports appear of the secrets that can be uncovered by setting learning algorithms loose on oceans of data. But there is no magic in machine learning, no hidden power, no alchemy. Instead there is an identifiable body of simple and practical techniques that can often extract useful information from raw data." [19]

Similarly, those who believe in data or information visualization are having a great time as the computer enables rapid display of large data sets with rich user control panels to support exploration [5]. Users can manipulate up to a million data items with 100-millisecond update of displays that present color-coded, size-coded markers for each item. With the right coding, human pre-attentive perceptual skills enable users to recognize patterns, spot outliers, identify gaps, and find clusters in a few hundred milliseconds. When data sets grow past a million items and cannot be easily seen on a computer display, users can extract relevant subsets, aggregate data into meaningful units, or randomly sample to create a manageable data set.

The commercial success of tools such as SAS JMP (www.sas.com), SPSS Diamond (www.spss.com), and Spotfire (www.spotfire.com) (Fig. 1), especially for pharmaceutical drug discovery and genomic data analysis, demonstrate the attraction of visualization. Other notable products include Inxight's Eureka (www.inxight.com) for multidimensional tabular data and Visual Insights' eBizinsights (www.visualinsights.com) for web log visualization.

Spence characterizes information visualization with this vignette "You are the owner of some numerical data which, you feel, is hiding some fundamental relation...you then glance at some visual presentation of that data and exclaim 'Ah ha! - now I understand.'"

[13]. But Spence also cautions that "information visualization is characterized by so many beautiful images that there is a danger of adopting a 'Gee Whiz' approach to its presentation."

5 A Spectrum of Discovery Tools

The happy resolution to these debates is to take the best insights from both extremes and create novel discovery tools for many different users and many different domains. Skilled problem solvers often combine observation at early stages, which leads to hypothesis-testing experiments. Alternatively they may have a precise hypothesis, but if they are careful observers during a controlled experiment, they may spot anomalies that lead to new hypotheses. Skilled problem solvers often combine statistical tests and visual presentation. A visual presentation of data may identify two clusters whose separate analysis can lead to useful results when a combined analysis would fail. Similarly, a visual presentation might show a parabola, which indicates a quadratic relationship between variables, but no relationship would be found if a linear correlation test were applied. Devotees of statistical methods often find that presenting their results visually helps to explain them and suggests further statistical tests.

The process of combining statistical methods with visualization tools will take some time because of the conflicting philosophies of the promoters. The famed statistician John Tukey (1915-2000) quickly recognized the power of combined approaches [14]: "As yet I know of no person or group that is taking nearly adequate, advantage of the graphical potentialities of the computer... In exploration they are going to be the data analyst's greatest single resource." The combined strength of visual data mining would enrich both approaches and enable more successful solutions [17]. However, most books on data mining have only brief discussion of information visualization and vice versa. Some researchers have begun to implement interactive visual approaches to data mining [10,2,15].

Accelerating the process of combining hypothesis testing with exploratory data analysis will also bring substantial benefits. New statistical tests and metrics for uniformity of distributions, outlier-ness, or cluster-ness will be helpful, especially if visual interfaces enable users to examine the distributions rapidly, change some parameters and get fresh metrics and corresponding visualizations.

6 Case Studies of Combining Visualization with Data Mining

One way to combine visual techniques with automated data mining is to provide support tools for users with both components. Users can then explore data with direct manipulation user interfaces that control information visualization components and apply statistical tests when something interesting appears. Alternatively, they can use data mining as a first pass and then examine the results visually. Direct manipulation strategies with user-controlled visualizations start with visual presentation of the world of action, which includes the objects of interest and the actions. Early examples included air traffic control and video games. In graphical user interfaces, direct manipulation means dragging files to folders or to the trashcan for deletion. Rapid incremental and reversible actions

encourage exploration and provide continuous feedback so users can see what they are doing. Good examples are moving or resizing a window. Modern applications of direct manipulation principles have led to information visualization tools that show hundreds of thousands of items on the screen at once. Sliders, check boxes, and radio buttons allow users to filter items dynamically with updates in less than 100 milliseconds.

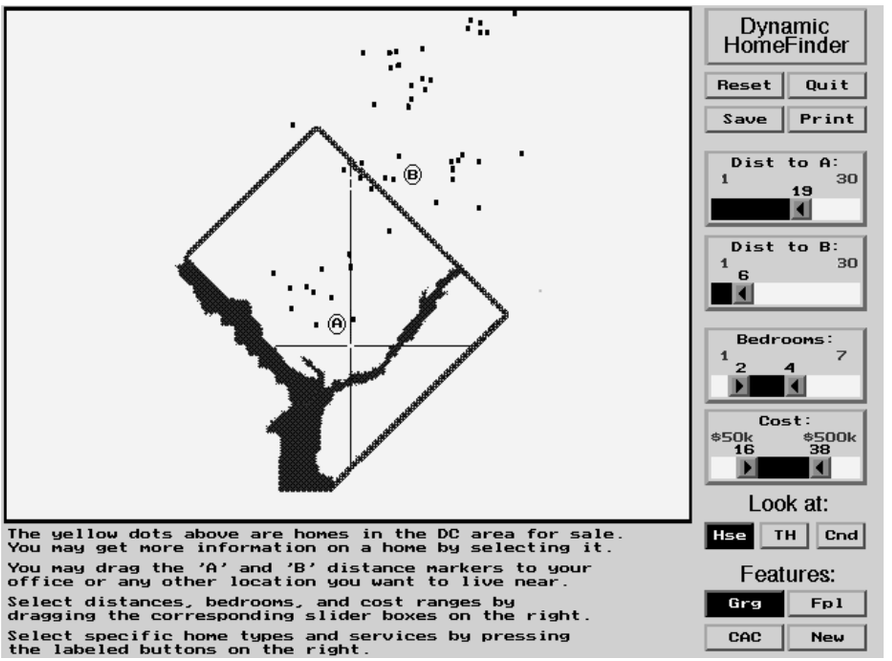


Fig. 2. Dynamic Queries HomeFinder with sliders to control the display of markers indicating homes for sale. Users can specify distances to markers, bedrooms, cost, type of house and features [18]

.

Early information visualizations included the Dynamic Queries HomeFinder (Fig. 2) which allowed users to select from a database of 1100 homes using sliders on home price, number of bedrooms, and distance from markers, plus buttons for other features such as fireplaces, central air conditioning, etc. [18].

This led to the FilmFinder [1] and then the successful commercial product, Spotfire (Fig. 1). One Spotfire feature is the View Tip that uses statistical data mining methods to suggest interesting pair-wise relationships by using linear correlation coefficients (Fig. 3). The ViewTip might be improved by giving more user control over the specification of interesting-ness that ranks the outcomes.

While some users may be interested in high linear correlation coefficients, others may be interested in low correlation coefficients, or might prefer rankings by quadratic,

exponential, sinusoidal or other correlations. Other choices might be to rank distributions by existing metrics such as skewness (negative or positive) or outlierness [3]. New metrics for degree of uniformity, cluster-ness, or gap-ness are excellent candidates for research. We are in the process of building a control panel that allows users to specify the distributions they are seeking by adjusting sliders and seeing how the rankings shift. Five algorithms have been written for 1-dimensional data and one for 2-dimensional data, but more will be prepared soon (Fig. 4).

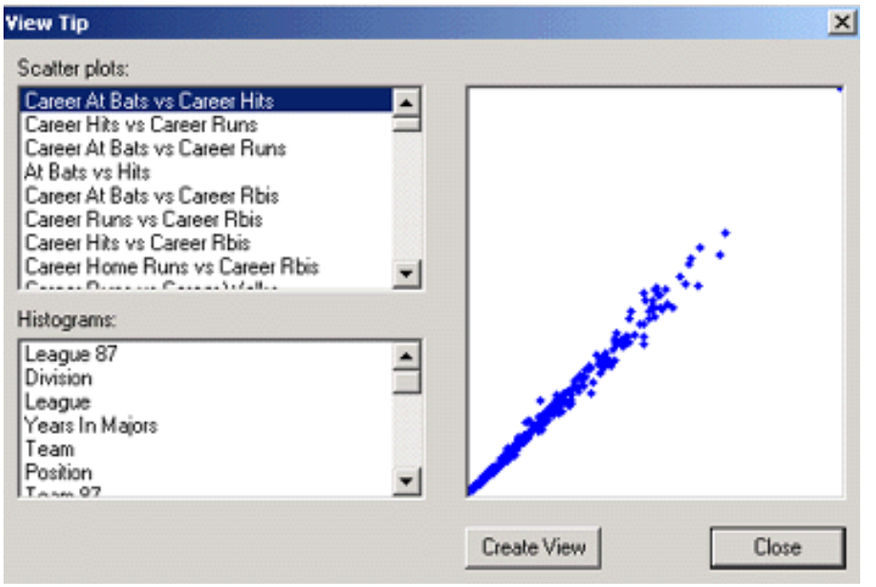


Fig. 3. Spotfire View Tip panel with ranking of possible 2-dimensional scatter plots in descending order by the strength of linear correlation. Here the strong correlation in baseball statistics is shown between Career At Bats and Career Hits. Notice the single outlier in the upper right corner, representing Pete Rose's long successful career.

A second case study is our work with time-series pattern finding [4]. Current tools for stock market or genomic expression data from DNA microarrays rely on clustering in multidimensional space, but a more user-controlled specification tool might enable analysts to carefully specify what they want [9]. Our efforts to build a tool, TimeSearcher, have relied on query specification by drawing boxes to indicate what ranges of values are desired for each time period (Fig. 5). It has more of the spirit of hypothesis testing. While this takes somewhat greater effort, it gives users greater control over the query results. Users can move the boxes around in a direct manipulation style and immediately see the new set of results. The opportunity for rapid exploration is dramatic and users can immediately see where matches are frequent and where they are rare.

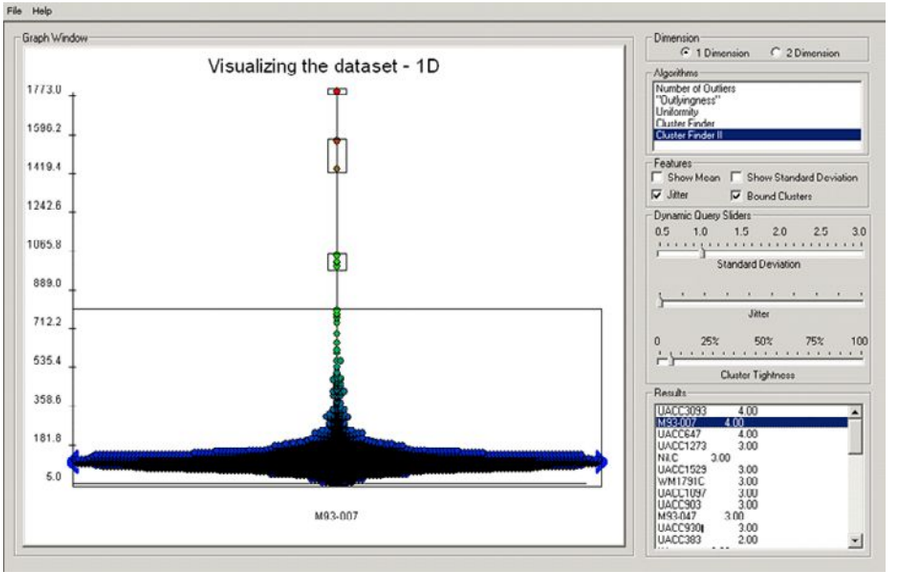


Fig. 4. Prototype panel to enable user specification of 1-dimensional distribution requirements. The user has chosen the Cluster Finder II in the Algorithm box at the top. The user has specified the cluster tightness desired in the middle section. The ranking of the Results at the bottom lists all distributions according to the number of identifiable clusters. The M93-007 data is the second one in the Results list and it has four identifiable clusters. (Implemented by Kartik Parija and Jaime Spacco).

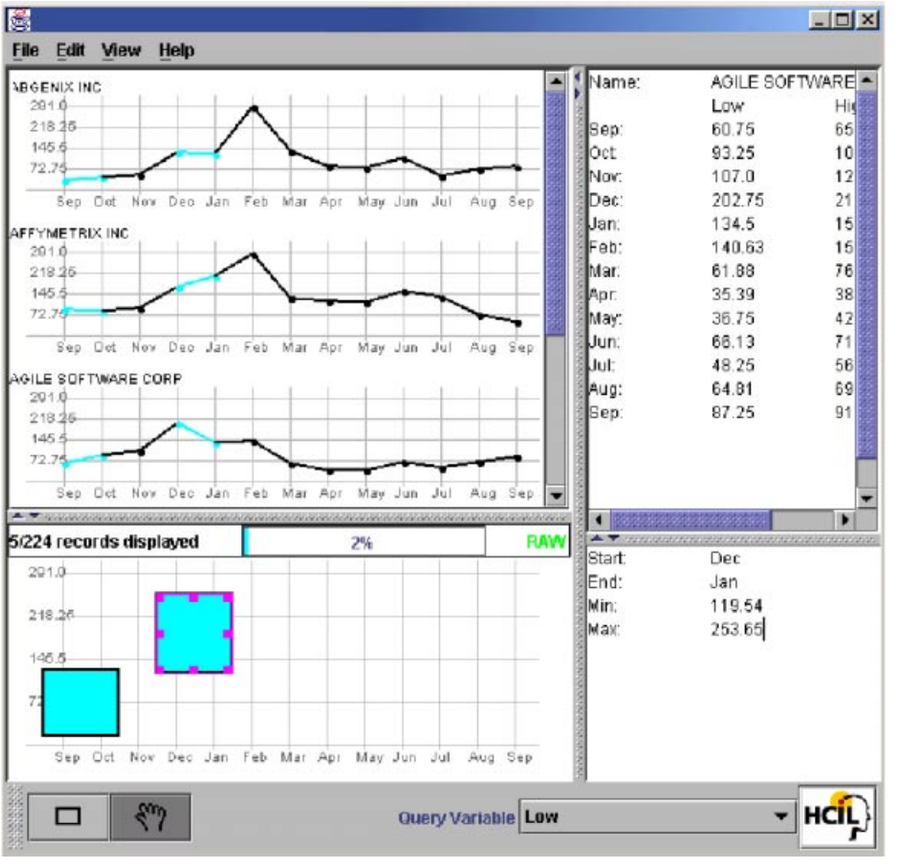


Fig. 5. TimeSearcher allows users to specify ranges for time-series data and immediately see the result set. In this case two timeboxes have been drawn and 5 of the 225 stocks match this pattern [9].

7 Conclusion and Recommendations

Computational tools for discovery, such as data mining and information visualization have advanced dramatically in recent years. Unfortunately, these tools have been developed by largely separate communities with different philosophies. Data mining and machine learning researchers tend to believe in the power of their statistical methods to identify interesting patterns without human intervention. Information visualization researchers tend to believe in the importance of user control by domain experts to produce useful visual presentations that provide unanticipated insights.

Recommendation 1: integrate data mining and information visualization to invent discovery tools. By adding visualization to data mining (such as presenting scattergrams to accompany induced rules), users will develop a deeper understanding of their data. By adding data mining to visualization (such as the Spotfire View Tip), users will be able to specify what they seek. Both communities of researchers emphasize exploratory data analysis over hypothesis testing. A middle ground of enabling users to structure their exploratory data analysis by applying their domain knowledge (such as limiting data mining algorithms to specific range values) may also be a source of innovative tools.

Recommendation 2: allow users to specify what they are seeking and what they find interesting. By allowing data mining and information visualization users to constrain and direct their tools, they may produce more rapid innovation. As in the Spotfire View Tip example, users could be given a control panel to indicate what kind of correlations or outliers they are looking for. As users test their hypotheses against the data, they find dead ends and discover new possibilities. Since discovery is a process, not a point event, keeping a history of user actions has a high payoff. Users should be able to save their state (data items and control panel settings), back up to previous states, and send their history to others.

Recommendation 3: recognize that users are situated in a social context. Researchers and practitioners rarely work alone. They need to gather data from multiple sources, consult with domain experts, pass on partial results to others, and then present their findings to colleagues and decision makers. Successful tools enable users to exchange data, ask for consultations from peers and mentors, and report results to others conveniently.

Recommendation 4: respect human responsibility when designing discovery tools. If tools are comprehensible, predictable and controllable, then users can develop mastery over their tools and experience satisfaction in accomplishing their work. They want to be able to take pride in their successes and they should be responsible for their failures. When tools become too complex or unpredictable, users will avoid their use because the tools are out of their control. Users often perform better when they understand and control what the computer does [11].

If complex statistical algorithms or visual presentations are not well understood by users they cannot act on the results with confidence. I believe that visibility of the statistical processes and outcomes minimizes the danger of misinterpretation and incorrect results. Comprehension of the algorithms behind the visualizations and the implications of layout encourage effective usage that leads to successful discovery.

Acknowledgements. Thanks to Mary Czerwinski, Lindley Darden, Harry Hochheiser, Jenny Preece, and Ian Witten for comments on drafts.

References

1. Ahlberg, C. and Shneiderman, B., Visual Information Seeking: Tight coupling of dynamic query filters with starfield displays, Proc. of ACM CHI '94 Human Factors in Computing Systems, ACM Press, New York (April 1994), 313-317 + color plates.
2. Ankerst, M., Ester, M., and Kriegel, H.-P., Towards an effective cooperation of the user and the computer for classification, Proc. 6th ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining, ACM, New York (2000), 179-188.
3. Barnett, Vic, and Lewis, Toby, Outliers in Statistical Data, John Wiley & Son Ltd; 3rd edition (April 1994).
4. Bradley, E., Time-series analysis, In Berthold, M. and Hand, E. (Editors), Intelligent Data Analysis: An Introduction, Springer (1999).
5. Card, S., Mackinlay, J, and Shneiderman, B. (Editors), Readings in Information Visualization: Using Vision to Think, Morgan Kaufmann Publishers, San Francisco, CA (1999).
6. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., (Editors), Advances in Knowledge Discovery and Data Mining. MIT Press, Cambridge, MA (1996).
7. Fisher, R.A., The Design of Experiments, Oliver and Boyd, Edinburgh (1935). 9th edition, Macmillan, New York (1971).
8. Han, Jiawei and Kamber, Micheline, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, San Francisco (2000).
9. Hochheiser, H. and Shneiderman, B., Interactive exploration of time-series data, In Proc. Discovery Science, Springer (2001).
10. Hinneburg, A., Keim, D., and Wawryniuk, M., HD-Eye: Visual mining of high-dimensional data, IEEE Computer Graphics and Applications 19, 5 (Sept/Oct 1999), 22-31.
11. Koenemann, J. and Belkin, N., A case for interaction: A study of interactive information retrieval behavior and effectiveness, Proc. CHI '96 Human Factors in Computing Systems, ACM Press, New York (1996), 205-212.
12. Montgomery, D., Design and Analysis of Experiments, 3rd ed, Wiley, New York (1991).
13. Spence, Robert, Information Visualization, Addison-Wesley, Essex, England (2001).
14. Tukey, John, The technical tools of statistics, American Statistician 19 (1965), 23-28. Available at: http://stat.bell-labs.com/who/tukey/memo/techtools.html
15. Ware, M., Frank, E., Homes, F., Hall, M., and Witten, I. H., Interactive machine learning: Letting users build classifiers, International Journal of Human-Computer Studies (2001, in press).
16. Weizenbaum, Joseph, Computer Power and Human Reason: From Judgment to Calculation, W. H. Freeman and Co., San Francisco, CA, (1976).
17. Westphal, Christopher and Blaxton, Teresa, Data Mining Solutions: Methods and Tools for Solving Real-World Problems, John Wiley & Sons (1999).
18. Williamson, Christopher, and Shneiderman, Ben, The Dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system, Proc. ACM SIGIR'92 Conference, ACM Press (1992), 338-346.
19. Witten, Ian, and Frank, Eibe, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann Publishers, San Francisco (2000).

Robot Baby 2001

Paul R. Cohen[1], Tim Oates[2], Niall Adams[3], and Carole R. Beal[4]

[1] Department of Computer Science, University of Massachusetts, Amherst
cohen@cs.umass.edu
[2] Department of Computer Science, University of Maryland, Baltimore County
oates@cs.umbc.edu
[3] Department of Mathematics, Imperial College, London
n.adams@ic.ac.uk
[4] Department of Psychology, University of Massachusetts, Amherst
cbeal@psych.umass.edu

Abstract. In this paper we claim that meaningful representations can be learned by programs, although today they are almost always designed by skilled engineers. We discuss several kinds of meaning that representations might have, and focus on a functional notion of meaning as appropriate for programs to learn. Specifically, a representation is meaningful if it incorporates an indicator of external conditions and if the indicator relation informs action. We survey methods for inducing kinds of representations we call structural abstractions. Prototypes of sensory time series are one kind of structural abstraction, and though they are not denoting or compositional, they do support planning. Deictic representations of objects and prototype representations of words enable a program to learn the denotational meanings of words. Finally, we discuss two algorithms designed to find the macroscopic structure of episodes in a domain-independent way.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, p. 29, 2001.

VML: A *View Modeling Language* for Computational Knowledge Discovery

Hideo Bannai[1], Yoshinori Tamada[2],
Osamu Maruyama[3], and Satoru Miyano[1]

[1] Human Genome Center, Institute of Medical Science, University of Tokyo
4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan.
{bannai,miyano}@ims.u-tokyo.ac.jp
[2] Department of Mathematical Sciences, Tokai University
1117 Kitakaname, Hiratuka-shi, Kanagawa 259-1292, Japan.
tamada@ss.u-tokai.ac.jp
[3] Faculty of Mathematics, Kyushu University
Kyushu University 36, Fukuoka 812-8581, Japan.
om@math.kyushu-u.ac.jp

Abstract. We present the concept of a functional programming language called *VML* (View Modeling Language), providing facilities to increase the efficiency of the iterative, trial-and-error cycle which frequently appears in any knowledge discovery process. In VML, functions can be specified so that returning values implicitly "remember", with a special internal representation, that it was calculated from the corresponding function. VML also provides facilities for "matching" the remembered representation so that one can easily obtain, from a given value, the functions and/or parameters used to create the value. Further, we describe, as VML programs, successful knowledge discovery tasks which we have actually experienced in the biological domain, and argue that computational knowledge discovery experiments can be efficiently developed and conducted using this language.

1 Introduction

The general flow and components which comprise the knowledge discovery process have come to be recognized [4,10] in the literature. According to these articles, the KDD process can be, in general, divided into several stages such as: data preparation (selection, preprocessing, transformation) data mining, hypothesis interpretation/evaluation, and knowledge consolidation. It is also well known that a typical process will not only go one-way through the steps, but will involve many feedback loops, due to the trial-and-error nature of knowledge discovery [2].

Most research in the literature concerning KDD focus on only a single stage of the process, such as the development of efficient and intelligent algorithms for a specific problem in the data mining stage. On the other hand, it seems that there has been comparatively little work which considers the process as a whole, concentrating on the *iterative* nature inherent in any KDD process.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 30–44, 2001.

More recently, the concept of *view* has been introduced for describing the steps of this process in a uniform manner [1,12,13,14]. Views are essentially functions over data. These functions, as well as their combinations, represent ways of looking at data, and the values they return are *attributes values* concerning their input arguments. The relationship between data, view, and the result obtained by applying a view to the data, can be considered as knowledge. The goal of KDD can be restated as the search for *meaningful* views. Views also provide an elegant interface for human intervention into the discovery process [1,12], whose need has been stressed in [9]. The iterative cycle of KDD consists very much of composing and decomposing of views, and facilities should be provided to assist these activities.

The purpose of this paper is to present the concept of a programming language, VML (View Modeling Language), which can help speed up this iterative cycle. We consider extending the Objective Caml (OCaml) language [27], a functional language which is a dialect of the ML [16] language. We chose a functional language for our base, since it can handle higher order values (functions) just like any other value, which should help in the manipulation of views. Also, functional languages have a reputation for enabling efficient and accurate programming of maintainable code, even for complex applications [6].

We focus on the fact that the primary difference between a view and a function, is that views must always have an interpretable *meaning*, because the knowledge must be interpretable to be of any use. The two extensions we consider are the keywords '`view`' and '`vmatch`'. '`view`' is used to bind a function to a name as well as instructing the program to *remember* any value resulting from the function. '`vmatch`' is a keyword for the *decomposing* of functional application, enabling the extraction of the origins of remembered values.

Of course, it is not impossible to accomplish the "remembering" with conventional languages. For example, we can have each function return a data structure which contains the resulting value and their representation. However, we wish to free the programmer from the labor of keeping track this data structure: what parameters were used where and when, by packaging this information implicitly into the language. As a result, the following tasks, for example, can be done without much extra effort:

- Interpret knowledge (functions and their parameters) obtained from learning/discovery programs.
- Reuse knowledge obtained from previous learning/discovery rounds.

Although we do not yet have a direct implementation of VML, we have been conducting computational experiments written in the C++ language based on the idea of views, obtaining substantial results [1,25]. We show how such experiments can be conducted comparatively easily by describing the experiments in terms of VML.

The structure of this paper is as follows: Basic concepts of views and VML is described in Section 3. We describe, using VML, two actual discovery tasks we have conducted in Section 4. We discuss various issues in Section 5.

2 Related Work

There have been several knowledge discovery systems which focus on similar problems concerning the KDD process as a whole. KEPLER [21] concentrates on the extensibility of the system, adopting a "plug-in architecture". CLEMENTINE [8] is a successful commercial application which focuses on human intervention, providing components which can be easily put together in many ways through a GUI interface. Our work is different and unique in that it tries to give a solution at a more generic level - until we understand the nature of the data, we must try, literally, any method we can come up with, and therefore *universality* is desired in our approaches.

Concerning the "remembering" of the origin of a value, one way to accomplish this is to remember the source code of the function. For example, some dialects of LISP provide a function called *get-lambda-expression*, which returns the actual lisp code of a given closure. However, this can return too much information concerning the value (e.g. the source code of a complicated algorithm). The idea in our work is to limit the information that the user will see, by regarding functions specified by the `view` keyword as the smallest unit of representation.

3 Concepts

In this section, we first briefly describe the concept of views, as found in [1]. Then, we discuss the basic concepts of VML, as an extension to the OCaml language [27], and give simple examples.

3.1 Entity, Views, and View Operation

Here, we review the definitions of entity, view, and view operation, and show how the KDD process can be described in terms of these concepts. An *entity set* E is a set of objects which may be distinguished from one another, representing the data under consideration. Each object $e \in E$ is called an *entity*. A *view* $v : E \to R$ is a function over E. v will take an entity e, and return some aspect (i.e. attribute value) concerning e. A *view operation* is an operation which generates new views from existing views and entities. Below are some examples:

Example 1. Given a view $v : E \to R$, a new view $v' : E \to R'$ may be created with a function $\psi : R \to R'$ (i.e. $v' \equiv \psi \circ v : E \xrightarrow{v} R \xrightarrow{\psi} R'$).

We can also consider n-ary functions as views. All arguments except for the argument expecting the entity can be regarded as *parameters* of the view.

Hypothesis generation via machine learning algorithms can also be considered as a form of view operation. The generated hypothesis can also be considered a view.

Example 2. Given a set of data records (entities) and their attributes (views), the ID3 algorithm [18] (view operator) generates a decision tree T. T is also a view

because it is a function which returns the class that a given entity is classified to. The generated view T can also be used as an input to other view operations, to create new views, which can be regarded as knowledge consolidation.

Views and view operators are combined to create new views. The structure of such combinations of a compound view, is called the *design* of the view. The task of KDD lies in the search for *good* views which explain the data. Knowledge concerning the data is encapsulated in its design. Human intervention can be conducted through the hand-crafted design of views by domain experts. To successfully assist the expert in the knowledge discovery process, the expert should be able to manipulate and understand the view design with ease.

3.2 Representations

Here, we describe the basic concepts in VML. We shall call *how* a certain value is created, its *representation*. For example, if an integer value 55 was created by adding the numbers from 1 to 10, the representation of 55 is informally, "add the integers from 1 to 10". A value may have multiple representations, but every representation should have only one corresponding value (except if there is some sort of random process in the representation). Intuitively, the representation for any value can be considered as the *source code* for computing that value. However, in VML, the representation is limited to only primitive values (first order values), and also *application* to functions specified with the `view` keyword, so that it is feasible for the users to understand and interpret the values, seeing only the information that they want to see.

The purpose of the `view` keyword is to specify that the runtime system should remember the representation of the return value of the function. We shall call such specified functions, *view functions*. Representations of values can be defined as:

```
rep ::= primv            (* primitive values *)
     |  vf rep1 ... repn (* application to view functions *)
     |  x . rep'         (* λ-abstraction of representations *)
```

`vmatch` is used to extract components from the representation of a value, by conducting pattern matching against the representation.

3.3 Simple Example

We give a simple example to illustrate basic OCaml syntax and the use of `view` and `vmatch` statements. The syntax and semantics of VML are the same as OCaml except for the added keywords. Only descriptions for the extended keywords are given, and the reader is requested to consult the Objective Caml Manual [27] for more information.

For the example in the previous subsection, a function which calculates the sum of positive integers 1 to n can be written in OCaml as:[1]

```
# let rec sumn n = if n <= 0 then 0 else (n + sumn (n-1));;
val sumn : int -> int = <fun>
# sumn 10;;                          (* apply 10 to sumn *)
- : int = 55
```

`let` binds the function (value) to the name `sumn`. `rec` specifies that the function is a *recursive* function (a function that calls itself). `n` is an argument of the function `sumn`. `int -> int` is the type of the function `sumn`, which reads as follows: "`sumn` is a function that takes a value of type `int` as an argument, and returns a value of type `int`". Notice that the type of `sumn` is *automatically inferred* by the compiler/interpreter, and need not be specified. Arguments can be applied to functions just by writing them consecutively.

The syntax of the `view` keyword is the same as the `let` statement. If we specify the above function with the `view` keyword in place of `let`: (we capitalize the first letter of view functions for convenience)

```
# view rec Sumn n = if n <= 0 then 0 else (n + Sumn (n-1));;
val Sumn : int -> int = <fun>::(n . Sumn n)
# Sumn 10;;
- : int = 55::(Sumn 10)
```

`Sumn` is defined as a view function, and therefore, values calculated from `Sumn` are implicitly remembered. In the above example, the return value is 55, and its representation, shown to the right of the double colon '::', is `(Sumn 10)`. We do not need to see the *inside* of `Sumn`, if we know the meaning of `Sumn`, to understand this value of 55.

The `vmatch` keyword is used to decompose a representation of a value and extract the function and/or any parameters which were used to created the value. Its syntax is the same as the `match` statement of OCaml, which is used for the pattern matching of miscellaneous data structures.

```
# let v = Sumn 10;; (* apply 10 to Sumn and bind the value to v *)
val v : int = 55::(Sumn 10)
# vmatch v with     (* Extract parameters used to calculate v *)
    (Sumn x) -> printf "%d was applied to Sumn\n" x
  | _ -> printf "Error: v did not match (Sumn x)\n";;
10 was applied to Sumn
- : unit = ()
```

In the above example, the representation of `v`, which is `(Sumn 10)`, is matched against the pattern `(Sumn x)`. If the match is successful, '`x`' in the pattern is

[1] The expressions starting after '`#`' and ending with '`;;`' is the input by the user, the others are responses from the compiler/interpreter. Comments are written between '`(*`' and '`*)`'.

assigned the corresponding value `10`. This value can be used in the expression to the right of '`->`' which is evaluated in case of a match. Multiple patterns matches can be attempted: each pattern and its corresponding expression are separated by '`|`', and the expression for the first matching pattern is evaluated. The underscore '_' represents a *wild card* pattern, matching any representation. The entire `vmatch` expression evaluates to the unit type `()` (similar to `void` in the C language) in this case, because `printf` is a function that executes a side-effect (print a string), and returns `()`.

3.4 Partial Application

Here, we consider how representations of partial applications to view functions can be done. We note, however, that our description here may contains subtle problems, for example, concerning the order of evaluation of the expressions, which may be counter intuitive when programs contain side-effects. A formal description and sample implementation resolving these issues can be found in [20].

In the previous examples, we added integers from 1 to n. Suppose we want to specify where to start also: add the integers from m to n. We can write the view function as follows:

```
# view rec Sum_m_to_n m n = if (m > n) then 0
                            else (n + (Sum_m_to_n m (n-1)));;
val Sum_m_to_n : int -> int -> int = <fun>::(m n . Sum_m_to_n m n)
# let sum3to = Sum_m_to_n 3;;          (* partial application *)
val sum3to : int -> int = <fun>::(n . Sum_m_to_n 3 n)
# sum3to 5;;
- : int = 12::(Sum_m_to_n 3 5)
```

`Sum_m_to_n` is a view function of type `int->int->int`, which can be read as "a function that takes two arguments of type `int` and returns a value of type `int`", *or*, "a function that takes one argument of type `int` and returns a value of type `int->int`". In defining `sum3to`, `Sum_m_to_n` is applied with only one argument, `3`, resulting in a function of type `int->int`. Applying another argument `5` to `sum3to` will result in the same value as `Sum_m_to_n 3 5`.

Partially applied values are matched as follows. Arguments not applied will only match the underscore '_':

```
# vmatch sum3to with
    (Sum_m_to_n x _)
      -> printf "Sum_m_to_n partially applied with %d\n" x
  | _ -> printf "failed match\n";;
Sum_m_to_n partially applied with 3
- : unit = ()
```

We can reverse the order of arguments by the `fun` keyword, which is essentially lambda abstraction.

```
# let sum10from = fun m -> Sum_m_to_n m 10;;
val sum10from : int -> int = <fun>::(m . Sum_m_to_n m 10)
# sum10from 5;;
- : int = 45::(Sum_m_to_n 5 10)
# vmatch sum10from with
    (Sum_m_to_n _ x)
      -> printf "Sum_m_to_n partially applied with %d\n" x
  | _ -> printf "failed match\n";;
Sum_m_to_n partially applied with 10
- : unit = ()
```

The representation for `sum10from` is the result obtained by β-reduction of the representation.

```
    (m . ((m n . Sum_m_to_n m n) m 10))
→β (m . ((n . Sum_m_to_n m n) 10))
→β (m . (Sum_m_to_n m 10))
```

3.5 Multiple Representations

In the example with `Sumn`, although `Sumn` recursively calls itself, the representation of the values generated in the recursive calls is not remembered, because the function `'+'` is not a view function. If multiple representations are to be remembered, they can be maintained with a list of representations, and `vmatch` will try to match any of the representations.

4 Actual Knowledge Discovery Tasks

We describe two computational knowledge discovery experiments, showing how VML can assist the programmer in such experiments. As noted in Section 1, VML is not yet fully implemented, and therefore the experiments conducted here were developed with the C++ language, using the HypothesisCreator library [25], based on the concept of views.

4.1 Detecting Gene Regulatory Sites

It is known that: for many genes, whether or not the gene expresses its function depends on specific proteins, called *transcription factors*, which bind to specific locations on the DNA, called *gene regulatory sites*. Gene regulatory sites are usually located in the upstream region of the coding sequence of the gene. Since proteins selectively bind to these sites, it is believed that common motifs exists for genes which are regulated by the same protein. We consider the case where the *2-block motif* model is preferred, that is, when the binding site cannot be characterized by a single motif, and 2 motifs should be searched for.

```
view ListDistAnd min max l1 l2: int->int->(int list)->(int list)->bool
  Return true if there exists e1 ∈ l1, e2 ∈ l2
  such that min ≤ (e2 − e1) ≤ max.
view AstrstrList mm pat str: astr_mismatch->string->string->(int list)
  Return the match positions (using approximate pattern matching)
  of a pattern as a list of int.
  The type astr_mismatch is the tuple (int * bool * bool * bool) where
  the int value is the maximum number of errors allowed, and the bool
  values are flags to permit the error types: insertion, deletion,
  and substitution, respectively.
```

Fig. 1. View functions used in the view design for detecting putative gene regulatory sites.

We develop a simple, original method, based on views. Testing the method on *B.subtilis* σ^A-dependent promoter sequences taken from [5], our method was able to rediscover the same results, as well as other candidates for 2-block motifs.

We started by modeling the 2-block motif for regulatory sites as consisting of three components: the motif pattern (a string pattern, with possible mismatches), the gap width of these patterns (how far apart they can be), and their positions (distance in base pairs from the beginning of the coding sequence). We construct a function with the following design (the representation is omitted):

```
# let orig pos len g_min g_max mm1 mm2 pat1 pat2 str =
    ListDistAnd g_min g_max
                (AstrstrList mm1 pat1 (Substring pos len str))
                (AstrstrList mm2 pat2 (Substring pos len str));;
val orig : int -> int -> float -> float -> astr_mismatch ->
    astr_mismatch -> string -> string -> string -> bool = <fun>
```

The explanations for the view functions used are given in Figure 1. The arguments except `str` are parameters, and when all the parameters are applied, a function of type `string->bool` is generated, returning `true` if a certain 2-block motif appears for a given string, and `false` otherwise. To look for good parameters, we take a supervised learning approach and randomly selected genes of *B.subtilis* not included in the original dataset, from the GenBank database [24], as negative data. The score of each view is based on its accuracy as a classification function that interprets whether or not an input sequence has the motifs. We looked at several top ranking views in order to evaluate them.

Numerous iterations with different search spaces yielded some interesting results. Selected results are shown in Figure 2. By limiting the search space by using knowledge obtained from previous work, we were able to come up with views `v1` and `v2` where the 2-block motifs were consistent or were the same with "TTGACA" and "TATAAT" as detected in [5,11]. We also ran the experiments with a wider range of parameters, and found a view `v3`, that could perfectly discriminate the positive and negative examples. Although a biological

```
v1: (str . ListDistAnd 20 30
              (AstrstrList (2,false,false,true) "ttgtca" (Substring -40 35 str))
              (AstrstrList (2,false,false,true) "tataat" (Substring -40 35 str)))
     true positive 102   false negative  40    =  71.8 %
    false positive   0    true negative 142    = 100.0 %
v2: (str . ListDistAnd 20 30
              (AstrstrList (2,false,false,true) "ttgaca" (Substring -40 35 str))
              (AstrstrList (2,false,false,true) "tataat" (Substring -40 35 str)))
     true positive 100   false negative  42    =  70.4 %
    false positive   0    true negative 142    = 100.0 %
v3: (str . ListDistAnd 25 35
              (AstrstrList (3,false,false,true) "atgatc" (Substring -50 65 str))
              (AstrstrList (2,false,false,true) "gttata" (Substring -50 65 str)))
     true positive 142   false negative   0    = 100.0 %
    false positive   0    true negative 142    = 100.0 %
```

Fig. 2. Representations of the results of our method to find regulatory sites.

interpretation must follow for the result to be meaningful, we were successful in finding a candidate for a novel result.

In this kind of experiment, VML can help the expert in the following way: Although the views are sorted by some score, it is difficult to check the validity of a view according to the score: i.e., a valuable view will probably have a high score, but a view with a high score may not be valuable. In the evaluation stage, there is a need for the expert to look at the many different views with adequately high scores, and see what kind of parameters were used to generate the view. This could be written easily in VML since it would be just to obtain and display the representations of high scoring functions.

4.2 Characterization of N-Terminal Sorting Signals of Proteins

Proteins are composed of amino acids, and can be regarded as strings consisting of an alphabet of 20 characters. Most proteins are first synthesized in the cytosol, and carried to specified locations, called *localization sites*. In most cases, the information determining the subcellular localization site is represented as a short amino acid sequence segment called a *protein sorting signal* [17]. Given an amino acid sequence, predicting where the protein will be carried to is an important and difficult problem in molecular biology. Although numerous signal sequences have been found, similarities between these sequence for the same localization site are not yet fully understood. Our aim was to come up with a predictor which could challenge TargetP [3], the state-of-the-art neural network based predictor, in terms of prediction accuracy while not sacrificing the *interpretability* of the classification rule.

Data available from the TargetP web-site [28] was used, consisting of 940 sequences containing 368 mTP (mitochondrial targeting peptides), 141 cTP (chloroplast transit peptides), 269 SP (signal peptides), and 162 “Other” sequences. The general approach was to: discuss with an expert on how to design the views, conduct computational experiments with those view designs, present results to the expert as feedback, and then repeat the process.

We first considered *binary* classifiers, which distinguishes sequences of a certain signal. The entity set is the set of amino acid sequences. The views we look for are of type `string -> bool`: for an amino sequence, return a Boolean value, `true` if the sequence contains a certain signal, and `false` if it does not. The views we designed (in time order) can be written in VML as follows (the meanings of each view function is given in Figure 3):

```
# let h1 pat mm ind pos len str =
    Astrstr mm pat (AlphInd ind (Substring pos len str));;
val h1 : string -> astr_mismatch -> (char -> char) -> int ->
    int -> string -> bool = <fun>::(pat mm ind pos len str .
    Astrstr mm pat (AlphInd ind (Substring pos len str))

# let h2 thr ind pos len str =
    GT (Average (AAindex ind (Substring pos len str))) thr;;
val h2 : float -> string -> int -> int -> string ->
    bool = <fun>::(thr ind pos len str .
    GT (Average (AAindex ind (Substring pos len str))) thr

# let h3 thr aaind pos1 len1 pat mm alphind pos2 len2 str =
    And (h1 pat mm alphind pos1 len1 str)
    (h2 thr aaind pos2 len2 str);;
val h3 : float -> string -> int -> int -> string ->
    astr_mismatch -> (char -> char) -> int -> int -> string ->
    bool = <fun>::(thr aaind pos1 len1 pat mm
    alphind pos2 len2 str . And (h1 pat mm alphind pos1 len1 str)
                                (h2 thr aaind pos2 len2 str)
```

Notice that after applying all the arguments except for the last string, we can obtain functions of type `string -> bool` as desired. For example, using view function `h2`, we can create a view function of type `string -> bool`:

```
# let f = h2 3.5 "BIGC670101" 5 20;;
val f : string -> bool =
        <fun>::(str .(GT (Average (AAindex "BIGC670101"
                                   (Substring 5 20 str)) 3.5)))
```

Each function is composed of view functions, so representation of such a function will contain information of the arguments. The representation of the above rule can be read as: "For a given amino acid sequence, first, look at the substring of length 20, starting from position 5. Then, calculate the average volume[2] of the amino acids appearing in the substring, and return true if it the value is greater than 3.5, false otherwise".

The task is now to find *good* parameters which defines a function that can accurately distinguish the signals. For each view design, a wide range of parameters were applied. For each combination of parameters and view design shown

[2] "BIGC670101" is the accession id for amino acid index: 'volume'.

```
view Substring pos len str : int -> int -> string -> string
  return substring: [pos,pos+len-1] of str. A negative value for pos
  means to count from the right end of the string.
view AlphInd ind str : (char -> char) -> string -> string
  convert str according to alphabet indexing ind. ind is a mapping of
  char->char, called an alphabet indexing[19], and can be considered
  as a classification of the characters of a given alphabet.
view Astrstr mm pat str : astr_mismatch -> string -> string -> bool
  approximate pattern matching[22]: match pat & str with mismatch mm.
  Type 'astr_mismatch' is explained in Figure 1.
view AAindex ac str : string -> string -> (float array)
  convert str to an array of float according to amino acid index: ac.
  ac is an accession id of an entry in the AAindex database[7]. Each
  entry in the database represents some biochemical property of amino
  acids, such as volume, hydropathy, etc., represented as a mapping of
  char -> float.
view Average v : float array -> float
  the average of the values in v
view GT x y : 'a -> 'a -> bool
  greater than
view And x y : bool -> bool -> bool
  Boolean 'and'
```

Fig. 3. View functions used in the view design to distinguish protein sorting signals.

above, we obtain a function: `string->bool`. The programmer need not worry about keeping track of the meanings of each function, because the representation may be consulted using the `vmatch` statement when needed. We apply all the protein sequences to this function, and calculate the score of this function as a classifier of a certain signal. Functions with the best scores are selected.

View design `h1`, looks for a pattern over a sequence converted by a classification of an alphabet [19]. We hoped to find some kind of structural similarities of the signals with this design, but we could not find satisfactory parameters which would let `h1` predict the signals accurately. Next, we designed a new view `h2` which uses the AAindex database [7], this time looking for characteristics of the amino acid composition of a sequence segment. This turned out to be very effective, especially for the SP set, and was used to distinguish SP from the other signals. For the remaining signals, we tried combining `h1` and `h2` into `h3`. This proved to be useful for distinguishing the "Other" set (those which do not have N-terminal signals), from mTP and cTP. We can see that the functional nature of VML enables the easy construction of the view designs.

By combining the views and parameters thus obtained for each signal type into a single decision list, we were able to create a rule which competes fairly well with TargetP in terms of prediction accuracy. The scores of a 5-fold cross-validation is shown in Table 1. The knowledge encapsulated in the view design

Table 1. The Prediction Accuracy of the Final Hypothesis (scores of TargetP [3] in parentheses) The score is defined by: $\frac{(tp \times tn - fp \times fn)}{\sqrt{(tp+fn)(tp+fp)(tn+fp)(tn+fn)}}$ where tp, tn, fp, fn are the number of true positive, true negative, false positive, and false negative, respectively (Matthews correlation coefficient (MCC) [15]).

True category	# of seqs	Predicted category				Sensitivity	MCC
		cTP	mTP	SP	Other		
cTP	141	96 (120)	26 (14)	0 (2)	19 (5)	0.68 (0.85)	0.64 (0.72)
mTP	368	25 (41)	309 (300)	4 (9)	30 (18)	0.84 (0.82)	0.75 (0.77)
SP	269	6 (2)	9 (7)	244 (245)	10 (15)	0.91 (0.91)	0.92 (0.90)
Other	162	8 (10)	17 (13)	2 (2)	135 (137)	0.83 (0.85)	0.71 (0.77)
Specificity		0.71 (0.69)	0.86 (0.90)	0.98 (0.96)	0.70 (0.78)		

was consistent with widely believed (but vague) characteristics of each signal, and the expert was surprised that such a simple rule could describe the sorting signals with such accuracy. A system called iPSORT was built based on these rules, and an experimental web service is provided at the iPSORT web-site [26].

`vmatch` can be useful in the following situation: After obtaining a good view of design `h2`, we may want to see if we can find a good view of design `h1`, but use the same substring sequence as `h2`. This can be regarded as first looking for a segment which has a distinct amino acid composition, and then looking closer at this segment, to see if structural characteristics of the segment can be found. This function can be written as:

```
# let newh f = vmatch f with
               GT (Average (AAindex _ (Substring p l _))) _ ->
                 fun pat mm ind str -> h1 pat mm ind p l str;;
val newh : '_a -> string -> astr_mismatch -> (char -> char)
           -> string -> bool = <fun>
```

If the representation of a function `h` was for example:

```
(str . GT (Average (AAindex ind (Substring 3 16 str))) 3.5)
```

then, the representation of `(newh h)` would become:

```
(pat mm ind str .
       (Astrstr mm pat (AlphInd ind (Substring 3 16 str))))
```

representing a function of design `h1`, but using the parameters of `h` of view design `h2` for `Substring`. Again, we need not worry about explicitly keeping track of what values were applied to `h2` to obtain `h`, since it is implicitly remembered and can be extracted by the `vmatch` keyword. Thus, we have seen that the design and manipulation of views can be done easily with VML, and would assist the trial-and-error cycle of the experiments.

5 Discussion

5.1 Implementation

In the C++ library, each view function is encapsulated in an instance of a class derived from the *view* class. The view class has a method for interpreting the value for an entity. Constructors for various derived classes can take other instances of view classes as arguments. The view class also has a method which returns the view classes which were used to build the instance (a facility for simulating vmatch, for decomposing the functions). However, after spending much time in development, we came to feel that C++ was error prone and rather tedious to code the view functions. Also, although the view classes encapsulate functions, the function itself could not be easily reused for other purposes.

For the points mentioned above, we can safely say that VML is advantageous over our C++ library. However, an efficient implementation of VML, which is beyond the scope of this paper, is a topic of interest. The implementation given in [20] uses the Camlp4 preprocessor (and printer) [23], which converts a VML program (with a different syntax from this paper) into an OCaml program, and it may be the case that there are optimizations that can be performed by a dedicated compiler.

5.2 Conclusion

We presented the concept of a language called VML, as an extension of the Objective Caml language. The advantages of VML are: 1) Since VML is a functional language, the composition and application of views can be done in a natural way, compared to imperative languages. 2) By defining the unit of knowledge as views, the programmer does not need to explicitly keep track of how each individual view was designed (i.e. manage data structures to remember the set of parameters). 3) The programmer can use "parts" of a good view which can only be determined perhaps at runtime, and apply it to another (the example in Section 4.2). 4) In an interactive interface, (i.e. a VML interactive interpreter), the user can compose and decompose views and view designs, and apply them to data. When the user accidently stumbles upon an interesting view, he/she can retrieve the design immediately.

Using VML, we modeled and described successful knowledge discovery tasks which we have actually experienced, and showed that the points noted above can lighten the burden of the programmer, and as a result, give way to speeding up the iterative trial-and-error cycle of computational knowledge discovery processes.

6 Acknowledgements

The authors would like to thank Sumii Eijiro of the University of Tokyo for his most valuable comments and suggestions.

This research was supported in part by Grant-in-Aid for Encouragement of Young Scientists and Grant-in-Aid for Scientific Research on Priority Areas (C) "Genome Information Science" from the Ministry of Education, Sports, Science and Technology of Japan, and the Research for the Future Program of the Japan Society for the Promotion of Science.

References

[1] H. Bannai, Y. Tamada, O. Maruyama, K. Nakai, and S. Miyano. Views: Fundamental building blocks in the process of knowledge discovery. In *Proceedings of the 14th International FLAIRS Conference*, pages 233–238. AAAI Press, 2001.

[2] P. Cheeseman and J. Stutz. Bayesian classification (AutoClass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*. AAAI Press/MIT Press, 1996.

[3] O. Emanuelsson, H. Nielsen, S. Brunak, and G. von Heijne. Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *J. Mol. Biol.*, 300(4):1005–1016, July 2000.

[4] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37–54, 1996.

[5] J. D. Helmann. Compilation and analysis of bacillus subtilis σ^A-dependent promoter sequences: evidence for extended contact between RNA polymerase and upstream promoter DNA. *Nucleic Acids Res.*, 23(13):2351–2360, 1995.

[6] J. Hughes. Why functional programming matters. *Computer Journal*, 32(2):98–107, 1989.

[7] S. Kawashima and M. Kanehisa. AAindex: Amino Acid index database. *Nucleic Acids Res.*, 28(1):374, 2000.

[8] T. Khabaza and C. Shearer. Data mining with Clementine. IEE Colloquium on 'Knowledge Discovery in Databases', 1995. *IEE Digest* No. 1995/021(B), London.

[9] P. Langley. The computer-aided discovery of scientific knowledge. In *Lecture Notes in Artificial Intelligence*, volume 1532, pages 25–39, 1998.

[10] P. Langley and H. A. Simon. Applications of machine learning and rule induction. *Communications of the ACM*, 38(11):54–64, 1995.

[11] X. Liu, D. L. Brutlag, and J. S. Liu. BioProspector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. In *Pacific Symposium on Biocomputing 2001*, volume 6, pages 127–138, 2001.

[12] O. Maruyama and S. Miyano. Design aspects of discovery systems. *IEICE Transactions on Information and Systems*, E83-D:61–70, 2000.

[13] O. Maruyama, T. Uchida, T. Shoudai, and S. Miyano. Toward genomic hypothesis creator: View designer for discovery. In *Discovery Science*, volume 1532 of *Lecture Notes in Artificial Intelligence*, pages 105–116, 1998.

[14] O. Maruyama, T. Uchida, K. L. Sim, and S. Miyano. Designing views in HypothesisCreator: System for assisting in discovery. In *Discovery Science*, volume 1721 of *Lecture Notes in Artificial Intelligence*, pages 115–127, 1999.

[15] B. W. Matthews. Comparison of predicted and observed secondary structure of t4 phage lysozyme. *Biochim. Biophys. Acta*, 405:442–451, 1975.

[16] R. Milner, M. Tofte, R. Harper, and D. MacQueen. *The Definition of Standard ML (Revised)*. MIT Press, 1997.

[17] K. Nakai. Protein sorting signals and prediction of subcellular localization. In P. Bork, editor, *Analysis of Amino Acid Sequences*, volume 54 of *Advances in Protein Chemistry*, pages 277–344. Academic Press, San Diego, 2000.

[18] J. Quinlan. Induction of decision trees. *Machine Learning 1*, 1:81–106, 1986.
[19] S. Shimozono. Alphabet indexing for approximating features of symbols. *Theor. Comput. Sci.*, 210:245–260, 1999.
[20] E. Sumii and H. Bannai. VMlambda: A functional calculus for scientific discovery. http://www.yl.is.s.u-tokyo.ac.jp/~sumii/pub/, 2001.
[21] S. Wrobel, D. Wettschereck, E. Sommer, and W. Emde. Extensibility in data mining systems. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 214–219, 1996.
[22] S. Wu and U. Manber. Fast text searching allowing errors. *Commun. ACM*, 35:83–91, 1992.
[23] Camlp4 - http://caml.inria.fr/camlp4/.
[24] GenBank - http://www.ncbi.nlm.nih.gov/Genbank.
[25] HypothesisCreator - http://www.hypothesiscreator.net/.
[26] iPSORT - http://www.hypothesiscreator.net/iPSORT/.
[27] Objective Caml - http://caml.inria.fr/ocaml/.
[28] TargetP - http://www.cbs.dtu.dk/services/TargetP/.

Computational Discovery of Communicable Knowledge: Symposium Report

Sašo Džeroski[1] and Pat Langley[2]

[1] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
Saso.Dzeroski@ijs.si, www-ai.ijs.si/SasoDzeroski/

[2] Institute for the Study of Learning and Expertise
2164 Staunton Court, Palo Alto, CA 94306 USA
langley@isle.org, www.isle.org/~langley/

Abstract. The *Symposium on Computational Discovery of Communicable Knowledge* was held from March 24 to 25, 2001, at Stanford University. Fifteen speakers reviewed recent advances in computational approaches to scientific discovery, focusing on their discovery tasks and the generated knowledge, rather than on the discovery algorithms themselves. Despite considerable variety in both tasks and methods, the talks were unified by a concern with the discovery of knowledge cast in formalisms used to communicate among scientists and engineers.

Computational research on scientific discovery has a long history within both artificial intelligence and cognitive science. Early efforts focused on reconstructing episodes from the history of science, but the past decade has seen similar techniques produce a variety of new scientific discoveries, many of them leading to publications in the relevant scientific literatures. Work in this paradigm has emphasized formalisms used to communicate among scientists, including numeric equations, structural models, and reaction pathways.

However, in recent years, research on data mining and knowledge discovery has produced another paradigm. Even when applied to scientific domains, this framework employs formalisms developed by artificial intelligence researchers themselves, such as decision trees, rule sets, and Bayesian networks. Although such methods can produce predictive models that are highly accurate, their outputs are not stated in terms familiar to scientists, and thus typically are not very communicable.

To highlight this distinction, Pat Langley organized the *Symposium on Computational Discovery of Communicable Knowledge*, which took place at Stanford University's Center for the Study of Language and Information on March 24 and 25, 2001. The meeting's aim was to bring together researchers who are pursuing computational approaches to the discovery of communicable knowledge and to review recent advances in this area. The primary focus was on discovery in scientific and engineering disciplines, where communication of knowledge is often a central concern.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 45–49, 2001.

Each of the 15 presentations emphasized the discovery tasks (the problem formulation and system input, including data and background knowledge) and the generated knowledge (the system output). Although artificial intelligence and machine learning traditionally focus on differences among algorithms, the meeting addressed the results of computational discovery at a more abstract level. In particular, it explored what methods for the computational discovery of communicable knowledge have in common, rather than the great diversity of methods used to that end.

The commonalities among methods for communicable knowledge discovery were summarized best by Raul Valdés-Pérez in a presentation titled *A Recipe for Designing Discovery Programs on Human Terms*. The key step in his recipe was identifying a set of possible solutions for some discovery task, as it is here that one can adopt a formalism that humans already use to represent knowledge. Valdés-Pérez viewed computational discovery as a problem-solving activity to which one can apply heuristic-search methods. He illustrated the recipe on the problem of discovering niche statements, i.e., properties of items that make them unique or distinctive in a given set of items.

The knowledge representation formalisms considered in the different presentations were diverse and ranged from equations through qualitative rules to reaction pathways. Most talks at the symposium fell within two broad categories. The first was concerned with equation discovery in either static systems or dynamic ones that change over time. The second addressed communicable knowledge discovery in biomedicine and in the related fields of biochemistry and molecular biology.

One formalism that scientists and engineers rely on heavily is equations. The task of equation discovery involves finding numeric or quantitative laws, expressed as one or more equations, from collections of measured numeric data. Most existing approaches to this problem deal with the discovery of algebraic equations, but recent work has also addressed the task of dynamic system identification, which involves discovering differential equations.

Takashi Washio from Osaka University presented a talk about *Conditions on Law Equations as Communicable Knowledge*, in which he discussed the conditions that equations must satisfy to be considered communicable. In addition to fitting the observed data, these include generic conditions and domain-dependent conditions. The former include objectiveness, generality, and reproducibility, as well as parsimony and mathematical admissibility with respect to unit dimensions and scale type constraints.

Kazumi Saito from Nippon Telegraph and Telephone and Mark Schwabacher from NASA Ames Research Center presented two related applications of computational equation discovery in the environmental sciences, both concerned with global models of the Earth ecosystem. Saito's talk on *Improving an Ecosystem Model Using Earth Science Data* addressed the task of revising an existing quantitative scientific model for predicting the net plant production of carbon in the light of new observations. Schwabacher's talk, *Discovering Communicable Scientific Knowledge from Spatio-Temporal Data in Earth Science*, dealt with

the problem of predicting from climate variables the Normalized Difference Vegetation Index, a measure of greenness and a key component of the previous ecosystem model.

Four presentations discussed the task of dynamic system identification, which involves identifying the laws that govern behavior of systems with continuous variables that change over time. Such laws typically take the form of differential equations. Two of these talks described extensions to equation discovery methods to address system identification, whereas the other talks reported work that began with methods for system identification and incorporated artificial intelligence techniques that take advantage of domain knowledge.

Saso Džeroski from the Jožef Stefan Institute, in his talk on *Discovering Ordinary and Partial Differential Equations*, gave an overview of computational methods for discovering both ordinary and partial differential equations, the second of which describe dynamic systems that involve change over several dimensions (e.g., space and time). Ljupčo Todorovski, from the same research center, discussed an approach that uses domain knowledge to aid the discovery process in his talk, *Using Background Knowledge in Differential Equations Discovery*. He showed how knowledge in the form of context-free grammars can constrain discovery in the domain of population dynamics.

Reinhard Stolle, from Xerox PARC, spoke about *Communicable Models and System Identification*. He described a discovery system that handles both structural identification and parameter estimation by integrating qualitative reasoning, numerical simulation, geometric reasoning, constraint reasoning, abstraction, and other mechanisms. Matthew Easley from the University of Colorado, Boulder, reported extensions to Stolle's framework in his presentation, *Incorporating Engineering Formalisms into Automated Model Builders*. His approach relied on input-output modeling to plan experiments and using the resulting data, combined with knowledge at different levels of abstraction, to construct a differential equation model.

The talk by Feng Zhao from Xerox PARC, *Structure Discovery from Massive Spatial Data Sets*, described an approach to analyzing spatio-temporal data that relies on the notion of spatial aggregation. This mechanism generates summary descriptions of the raw data, which it characterizes at varying levels of detail. Zhao reported applications to several challenging problems, including the interpretation of weather data, optimization for distributed control, and the analysis of spatio-temporal diffusion-reaction patterns.

The rapid growth of biological databases, such as that for the human genome, has led to increased interest in applying computational discovery to biomedicine and related fields. Five presentations at the symposium focused on this general area. They covered a variety of discovery methods, including both propositional and first-order rule induction, genetic programming, theory revision, and abductive inference, with similar breadth in the biological discovery tasks to which they were applied.

Bruce Buchanan and Joseph Phillips, from the University of Pittsburgh, gave a presentation titled *Introducing Semantics into Machine Learning*. This focused

on their incorporation of domain knowledge into rule-induction algorithms to let them find interesting and novel relations in medicine and science. They reviewed both syntactic and semantic constraints on the rule discovery process and showed that stronger forms of background knowledge increase the chances that discovered rules are understandable, interesting, and novel.

Stephen Muggleton from York University, in his talk *Knowledge Discovery in Biological and Chemical Domains*, described his application of first-order rule induction to predicting the structure of proteins, modeling the relations between a chemical's structure and its activity, and predicting a protein's function from its structure (e.g., identifying precursors of neuropeptides). Knowledge discovered in these efforts has appeared in journals for the respective scientific areas.

John Koza from Stanford University presented *Reverse Engineering and Automatic Synthesis of Metabolic Pathways from Observed Data*. His approach utilized genetic programming to carry out search through a space of metabolic pathway models, with search directed by the models' abilities to fit time-series data on observed chemical concentrations. The target model included an internal feedback loop, a bifurcation point, and an accumulation point, suggesting the method can handle complex metabolic processes.

The presentation by Pat Langley, from the Institute for the Study of Learning and Expertise, addressed *Knowledge and Data in Computational Biological Discovery*. He reported an approach that used data on gene expressions to revise a model of photosynthetic regulation in Cyanobacteria previously developed by plant biologists. The result was an improved model with altered processes that better explains the expression levels observed over time. The ultimate goal is an interactive system to support human biologists in their discovery activities.

Marc Weeber from the U.S. National Library of Medicine reported on a quite different approach in his talk on *Literature-based Discovery in Biomedicine*. The main idea relies on utilizing bibliographic databases to uncover indirect but plausible connections between disconnected bodies of scientific knowledge. He illustrated this method with a successful example of finding potentially new therapeutic applications for an existing drug, thalidomide.

Sakir Kocabas, from Istanbul Technical University, talked about *The Role of Completeness in Particle Physics Discoveries*, which dealt with a completely different domain. He described a computational model of historical discovery in particle physics that relies on two main criteria – consistency and completeness – to postulate new quantum properties, determine those properties' values, propose new particles, and predict reactions among particles. Kocabas' system successfully simulated an extended period in the history of this field, including discovery of the neutrino and postulation of the baryon number.

At the close of the symposium, Lorenzo Magnani from the University of Pavia commented on the presentations from a philosophical viewpoint. In particular, he cast the various efforts in terms of his general framework for abduction, which incorporates different types of explanatory reasoning. The gathering also spent time honoring the memory of Herbert Simon and Jan Żytkow, both of whom played seminal roles in the field of computational scientific discovery.

Further information on the symposium is available at the World Wide Web page `http://www.isle.org/symposia/comdisc.html`. This includes information about the speakers, abstracts of the presentations, and pointers to publications related to their talks. Slides from the presentations can be found at the Web page `http://math.nist.gov/~JDevaney/CommKnow/`. Sašo Džeroski and Ljupčo Todorovski are currently editing a book based on the talks given at the symposium. Information on the book will appear at the symposium page and the first author's Web page as it becomes available.

Acknowledgements

The Symposium on Computational Discovery of Communicable Knowledge was supported by Grant NAG 2-1335 from NASA Ames Research Center and by the Nippon Telegraph and Telephone Corporation.

References

Bradley, E., Easley, M., & Stolle, R. (in press). Reasoning about nonlinear system identification. *Artificial Intelligence*.

Kocabas, S., & Langley, P. (in press). An integrated framework for extended discovery in particle physics. *Proceedings of the Fourth International Conference on Discovery Science*. Washington, D.C.: Springer.

Koza, J. R., Mydlowec, W., Lanza, G., Yu, J., & Keane, M. A. (2001). Reverse engineering and automatic synthesis of metabolic pathways from observed data using genetic programming. *Pacific Symposium on Biocomputing*, *6*, 434–445.

Lee, Y., Buchanan, B. G., & Aronis, J. M. (1998). Knowledge-based learning in exploratory science: Learning rules to predict rodent carcinogenicity. *Machine Learning*, *30*, 217–240.

Muggleton, S. (1999). Scientific knowledge discovery using inductive logic programming. *Communications of the ACM*, *42*, 42–46.

Saito, K., Langley, P., Grenager, T., Potter, C., Torregrosa, A., & Klooster, S. A. (in press). Computational revision of quantitative scientific models. *Proceedings of the Fourth International Conference on Discovery Science*. Washington, D.C.: Springer.

Schwabacher, M., & Langley, P. (2001). Discovering communicable scientific knowledge from spatio-temporal data. *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 489–496). Williamstown, MA: Morgan Kaufmann.

Shrager, J., Langley, P., & Pohorille, A. (2001). *Guiding revision of regulatory models with expression data.* Unpublished manuscript, Institute for the Study of Learning and Expertise, Palo Alto, CA.

Todorovski, L., & Džeroski, S. (2000). Discovering the structure of partial differential equations from example behavior. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 991–998). San Francisco: Morgan Kaufmann.

Valdés-Pérez, R. E. (1999). Principles of human-computer collaboration for knowledge discovery in science. *Artificial Intelligence*, *107*, 335–346.

Washio, T., Motoda, H., & Niwa, Y. (2000). Enhancing the plausibility of law equation discovery. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 1127–1134). San Francisco: Morgan Kaufmann.

Yip, K., & Zhao, F. (1996). Spatial aggregation: Theory and applications. *Journal of Artificial Intelligence Research*, *5*, 1–26.

Bounding Negative Information in Frequent Sets Algorithms*

I. Fortes[1], J.L. Balcázar[2], and R. Morales[3]

[1] Dept. Applied Mathematic, E.T.S.I. Informática, Univ. Málaga. Campus Teatinos. 29071 Málaga, Spain
ifortes@ctima.uma.es
[2] Dept. LSI, Univ. Politècnica de Catalunya. Campus Nord. 08034 Barcelona, Spain
balqui@lsi.upc.es
[3] Dept. Languages and Computer Science, E.T.S.I. Informática, Univ. Málaga. Campus Teatinos. 29071 Málaga, Spain
morales@lcc.uma.es

Abstract. In Data Mining applications of the frequent sets problem, such as finding association rules, a commonly used generalization is to see each transaction as the characteristic function of the corresponding itemset. This allows one to find also correlations between items not being in the transactions; but this may lead to the risk of a large and hard to interpret output. We propose a bottom-up algorithm in which the exploration of facts corresponding to items not being in the transactions is delayed with respect to positive information of items being in the transactions. This allows the user to dose the association rules found in terms of the amount of correlation allowed between absences of items. The algorithm takes advantage of the relationships between the corresponding frequencies of such itemsets. With a slight modification, our algorithm can be used as well to find all frequent itemsets consisting of an arbitrary number of present positive attributes and at most a predetermined number k of present negative attributes.

* Work supported in part by the EU ESPRIT IST-1999-14186 (ALCOM-FT), EU EP27150 (NeuroColt), CIRIT 1997SGR-00366 and PB98-0937-C04 (FRESCO).

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 50–58, 2001.

1 Introduction

Data Mining or Knowledge Discovery in Databases (KDD) is a field of increasing interest with strong connections with several research areas such as databases, machine learning, and statistics. It aims at finding useful information from large masses of data; see [5]. One of the most relevant subroutines in applications of this field is finding frequent itemsets within the transactions in the database. This task consists of finding highly frequent itemsets, by comparing their frequency of occurrence within the given database with a given parameter σ. This problem can be solved by the well-known Apriori algorithm [2].

The Apriori algorithm is a method of searching the lattice of itemsets with respect to itemset inclusion. The strategy starts from the empty set and scans itemsets from smaller to larger in an incremental manner. The Apriori algorithm uses this strategy to effectively prune away a substantial number of unproductive itemsets.

The frequent sets that result from this task can be used then to discover association rules that have support and confidence values no smaller than the user-specified minimum thresholds [1], or to solve other related Knowledge Discovery problems [7]. We do not discuss here how to form association rules from frequent itemsets, nor any other application of these; but focus on the performance of that very step, finding highly frequent patterns, whose complexity dominates by far the computational cost of many such applications.

Here we considered the case where each transaction of the database is a binary-valued function of the attributes. The difference with the itemsets view is that now we look for patterns where the non-occurrence of an item is important too. This is formalized in terms of partial functions, which, on each item, may include it (value 1), exclude it (value 0), or not to consider it (undefined).

It was noticed in [6] that essentially the same algorithms, with the same "a priori" pruning strategies, can be applied to many other settings in which one looks for a certain theory on a certain formal language according to a certain predicate that is monotone on a generalization/specialization relation. In particular, our setting with binary-valued attributes falls into this category, and actually there exist implementations of the Apriori algorithm that solve the problem for the setting where each transaction is actually a function. Thus, they can be used to solve the problem of finding partial functions whose frequency is over some threshold.

However, it is known that direct use of these algorithms on real life data frequently come up with extremely large numbers of frequent sets consisting "only of zeros"; for example, in the prototypical case of market basket data, certainly the number of items is overwhelmingly larger than the average number of items bought, and this means that the output of any frequent sets algorithm will contain large amounts of information of the sort "most of the times that scotch is not bought, bourbon is not bought either, with large support". If such negative information is not desired at all, the original Apriori version can be used; but there may be cases where limited amounts of negative information are deemed useful, for instance looking for alternative products that can act

as mutual replacements, and yet one does not want to be forced into a search through the huge space of all partial functions. We are interested in producing algorithms that will provide frequent "itemsets" that have "missing" products, but in a controlled manner, so that they are useful when having some missing products in the itemsets is important but not so much as the products that are in the itemsets.

Here we develop a variant of the Apriori algorithm that, if supplied with a limit k on the maximum number of negated attributes desired in the output frequent sets, will take advantage of this fact, and produce frequent itemsets for which this limit is obeyed. Of course, it does so in a much more efficient way than just applying Apriori and discarding the part of the output that does not fulfill this condition. First, because the exploration is organized in a way that naturally reflects the condition on the output. Second, because we know that items may be, or not be, in each itemset, but not both implies complementarity relationships between the frequencies of "itemsets" that contain, or do not contain, a given item. We use these relationships to find out frequencies of some "itemsets" without actually counting them, thus saving computational work.

2 Preliminaries

Now, we give the concepts that we will use along the paper. We consider a database $\mathcal{T} = \{t_1, \ldots, t_N\}$ with N rows over a set $R = \{A_1, \ldots, A_n\} = \{A_i : i \in I\}$ of binary-valued attributes, that can be seen as either items or columns; actually they just serve as a visual aid for their index set $I = \{1, \ldots, n\}$.

Each row, or transaction, maps R into $\{0, 1\}$. For $A \in R$, we also write $A \in t_l$ for $t_l(A) = 1$ and, departing from standard use, $\overline{A} \in t_l$ for $t_l(A) = 0$. Obviously, $A \in t_l$ or $\overline{A} \in t_l$ but not both. The database is actually a multiset of transactions. Each transaction has a unique identifier.

As for partial functions, they map a subset of R into $\{0, 1\}$; those that are defined for exactly ℓ attributes are called ℓ-itemsets. The goal of our algorithm will be to find frequent itemsets with any number of attributes mapped to 0 and any number of attributes mapped to 1; but in some specific order. Our notation for these partial functions is as follows. For $p \in \mathcal{P}(I)$ and $s \in \mathcal{P}(I - p)$, $(s \cap p = \emptyset)$ we denote the subset $A^{p,s}$ and identify it with the partial function mapping the subset $A^p = \{A_i : i \in p\}$ to 1, the subset $A^s = \{A_j : j \in s\}$ to 0 and undefined on the rest. Itemsets $A^{p,s}$ are called k-negative itemsets where $|s| = k,\ k = 0, \ldots, n$. If $|s| = 0$ then we have the positive itemset $A^{p,\emptyset}$.

We identify partial functions defined on a single attribute A_j, namely, $A^{\{j\},\emptyset}$ or $A^{\emptyset,\{j\}}$, with the corresponding symbol A_j or $\overline{A_j}$ respectively. A transaction can be seen as a total function. An itemset can be seen as a partial function. If the partial function can be extended to the total function corresponding to a transaction then we say that an itemset is a subset of a transaction and we employ the standard symbol $\subseteq$ for this case.

The support of an itemset (or partial function) is defined as follows.

Definition 1. *Let $R = \{A_1, \ldots, A_n\} = \{A_i : i \in I\}$ be a set of n items and let $\mathcal{T} = \{t_1, \ldots, t_N\}$ be a database of transactions as before. The support or frequency of an itemset A is the ratio of the number of transactions on which it occurs as a subset to the total number of transactions. Therefore:*

$$fr(A) = \frac{|\{t \in \mathcal{T} : A \subseteq t\}|}{N}$$

Given a user-specified minimum support value (denoted by σ), we say than an itemset A is *frequent* if its support is more than the minimum support, i.e. $fr(A) \geq \sigma$.

We introduce a natural structure in the itemset space by placing them into "floors" and "levels". The floor k contains the itemsets with k negative attributes. In each floor, the itemsets are organized in levels (as usual): the level is the number of the attributes of the itemset. Thus, in floor zero we place positive itemsets, ordered by itemset inclusion (or equivalently, index set inclusion); in the first floor we place all itemsets with one attribute valued to 0, organized similarly, and related similarly to the itemsets in floor zero. In floor k we place all the itemsets with k attributes valued to 0, organized levelwise in the standard way, and related similarly to the itemsets in other floors.

Thus we are considering the order relation defined as follows:

Definition 2. *For $p \in \mathcal{P}(I)$, $s \in \mathcal{P}(I-p)$, $q \in \mathcal{P}(I)$, and $t \in \mathcal{P}(I-q)$, given partial functions $X = A^{p,s}$ and $Y = A^{q,t}$, we denote by $X \preccurlyeq Y$ the fact that $p \subseteq q$ and $s \subseteq t$.*

With respect to this relation, the property of having frequency larger than any threshold is antimonotone, since $X \preccurlyeq Y$ implies $fr(X) \geq fr(Y)$. Thus, whenever an itemset is not frequent enough, neither is any of its extensions, and this fact allows one to prune away a substantial number of unproductive itemsets. Therefore, frequent sets algorithms can be applied rather directly to this case. Our purpose now is to aim at a somewhat more refined algorithm.

Now, we give a simple example to show the structure of the itemset space. This example will be useful to describe the frequent itemset candidate generation and the path that follows our algorithm for it.

Example: Let $R = \{A, B, C, D\}$ be the set of four items. In this case, we use four floors to represent the itemsets with any number of negative attributes and any number of positive attributes. In each rectangle, the pair (f, ℓ) indicates the floor f (number of negative attributes in the itemsets of this rectangle) and level ℓ (cardinality of the itemsets of this rectangle). See figure 1.

3 Algorithm Bounded-neg-Apriori

Our algorithm performs the same computations as Apriori on the zero floor, but then uses the frequencies computed to try to reduce the computational effort spent on 1-negative itemsets. This process goes on along all floors. Overall,

$ABCD$ (0,4)	$AB\overline{C}D$,..... (1,4)	$\overline{A}\overline{B}CD$,..... (2,4)	$\overline{A}\overline{B}\overline{C}D$,..... (3,4)	$\overline{A}\overline{B}\overline{C}\overline{D}$ (4,4)
ABC, BCD,..... (0,3)	$A\overline{B}D$,..... (1,3)	$\overline{A}\overline{B}D$,..... (2,3)	$\overline{B}\overline{C}D$,..... (3,3)	
AB, BC,CD,..... (0,2)	$A\overline{B}$,.... (1,2)	$\overline{B}\overline{D}$,... (2,2)		
A,B,C,D (0,1)	$\overline{A},\overline{B},\overline{C},\overline{D}$ (1,1)			
$\emptyset$ (0,0)				

Fig. 1. The structure of the itemset space

bounded-neg-Apriori can be seen as a refinemet of Apriori in which the explicit evaluation of the frequency of k-negative itemsets is avoided, since it can be obtained from some itemsets of the previous floor, if they are processed in the appropriate order.

We use a number of very easy properties of the frequencies. Of course all of the frequencies are real numbers in $[0, 1]$.

Proposition 1. *Let $p \in \mathcal{P}(I)$ be arbitrary, and $s \in \mathcal{P}(I-p)$ with $|s| \geq 1$.*

1. *For each $j \in s$, $fr(A^{p,s}) = fr(A^{p,s-\{j\}}) - fr(A^{p\cup\{j\},s-\{j\}})$ and, $fr(A^{\emptyset,\emptyset}) = 1$*
2. *$A^{p,s}$ is frequent iff $\exists j \in s$, $fr(A^{p,s-\{j\}}) > \sigma + fr(A^{p\cup\{j\},s-\{j\}})$.*

Remark 1: Each of the up to $|s|$-many ways of decomposing $fr(A^{p,s})$ in part 1 leads to the same result: if $fr(A^{p,s-\{j\}}) < \sigma$, for any $j \in s$, then $A^{p,s}$ is not frequent.

We will also use the following easy properties regarding the relation of the threshold σ to the value one-half. They allow for some extra pruning to be done for quite high frequency values (although this case might be infrequently occurring in practice).

Proposition 2. *Let $p \in \mathcal{P}(I)$ be arbitrary, and $s \in \mathcal{P}(I-p)$, arbitrary for statements not depending on p.*

1. *$|fr(A_j) - 0.5| < |\sigma - 0.5| \Leftrightarrow |fr(\overline{A}_j) - 0.5| < |\sigma - 0.5|$.*
2. *If $\sigma < 0.5$ then $fr(A_j) \leq \sigma \Rightarrow fr(\overline{A}_j) > \sigma$ and $fr(A_j) > 1 - \sigma \Leftrightarrow fr(\overline{A}_j) < \sigma$.*
3. *If $\sigma > 0.5$ then $fr(A_j) \geq \sigma \Rightarrow fr(\overline{A}_j) < \sigma$ and $fr(A_j) < 1 - \sigma \Leftrightarrow fr(\overline{A}_j) > \sigma$.*
4. *$\forall j \in s$, if $\sigma > 0.5$ and $fr(A^{p,s-\{j\}}) > \sigma + fr(A^{p\cup\{j\},s-\{j\}})$ then $fr(A^{p\cup\{j\},s-\{j\}}) < \sigma$, i.e. in this case $A^{p\cup\{j\},s-\{j\}}$ is not frequent.*

Remark 2 If $\sigma > 0.5$ and $\exists j \in s \,/\, fr(A^{p\cup\{j\},s-\{j\}}) > \sigma$ then $A^{p,s}$ is not frequent.

3.1 Candidate Generation

Moving to the next round of candidates once all frequent ℓ-itemsets have been identified corresponds to moving up, in all possible ways, one step within the same floor, and climbing up in all possible ways to the next floor.

More formally, at the floor zero, frequent set $A^{p,\emptyset}$ leads to consideration as potential candidates of the following itemsets: all $A^{q,\emptyset}$ where $q = p \cup \{i\}$ and all $A^{p,\{j\}}$, for $j \notin p$. Also, itemset $A^{p,\{j\}}$ would lead to $A^{q,\{j\}}$ for $q = p \cup \{i\}$, for $i \notin p$ and $i \neq j$; our algorithm does not use this last sort of steps.

In the other floors the movements are in the same form. For all $p \in \mathcal{P}(I)$ and $s \neq \emptyset$, from $A^{p,s}$ we can climb up to the next floor to $A^{p,t}$ where $t = s \cup \{j\}$, for $j \in \mathcal{P}(I - \{s \cup p\})$. Also, itemset $A^{p,s}$ would lead to $A^{q,s}$ for $q = p \cup \{i\}$, for $i \notin p$ and $i \notin s$ but we will not use such steps either.

Therefore the scheme of the search of frequent itemsets with k 0-valued attributes (i.e. in the floor k) is based on the following: whenever enough frequencies in the previous floor are known to test it, if $fr(A^{p,s-\{j\}}) > \sigma + fr(A^{p\cup\{j\},s-\{j\}})$ where $j \in s$, then we know $fr(A^{p,s}) > \sigma$ so that it can be declared frequent; moreover, for $\sigma > 0.5$ this has to be tested only when that $A^{p\cup\{j\},s-\{j\}}$ turned out to be nonfrequent although $A^{p,s-\{j\}}$ was frequent.

Example: Let us turn our atention again to the example. Let us suppose that $\sigma < 0.5$; we explain the process of candidate generation and the path that our algorithm follows for it. Suppose that the maximal itemsets to be found are ABC, $AB\overline{C}$, and $A\overline{B}$. Thus, A, B, C are frequent items, and also $\overline{B}$ and $\overline{C}$ are frequent 'negative items'. At the initialization, we find that D, $\overline{A}$, and $\overline{D}$ cannot appear in any frequent itemset. The algorithm stores this information by means of the set I (defined later). In the following step, we take into consideration as potential candidates, firstly the itemsets in $(0, 2)$, secondly in $(1, 2)$, and at last, in $(2, 2)$ that verify the conditions. There we find the frequent itemsets are AB, AC, BC, $A\overline{B}$, $A\overline{C}$, $B\overline{C}$. At this moment, we know that there do not exist frequent itemsets in $(2, 2)$. So, there will not exist frequent itemsets in (f, ℓ) with $f \geq 2$, $\ell > 2$ and $\ell \geq f$. This information is used in the algorithm by means of the set J (defined later) to refine the search of candidate generation. In the following step we scan for frequent itemsets in $(0, 3)$ and $(1, 3)$ and ABC, $AB\overline{C}$ are frequent itemsets, and the exploration of the next level proves that, together with $A\overline{B}$, they are the maximal frequent itemsets. Along the example it is clear how the algorithm would proceed in case we are given a bound on the number of negative attributes present: this would just discard floors that do not obey that limitation.

3.2 The Algorithm

Now, we present the algorithm in a more precise form. The algorithm has as input the set of attributes, the database, and the threshold σ on the support. The output of the algorithm is the set of all frequent itemsets with negative and positive itemsets. Also, a similar algorithm can be easily developed to find the

set of all frequent itemsets with at most k negative attributes: simply impose explicitly the bound k on the corresponding loop in the algorithm.

Let us consider the symbol f for the floor (that is the number of negative attributes of the itemset, $0 \leq f \leq n$) and the symbol ℓ for the level (the number of the attributes of the itemset $0 \leq \ell \leq n$): we will write the sets $C_{f,\ell}$ and $L_{f,\ell}$ for candidates and frequent itemsets respectively. At the beginning we suppose that all $C_{f,\ell}$ and $L_{f,\ell}$ for $f \leq \ell \leq n$ are empty.

With respect to this notation our algorithm traces the following path: $(0,1),(1,1);(0,2),(1,2),(2,2);(0,3),(1,3),(2,3),(3,3);$, etc (recall to the example).

Now, we present the algorithm in a pseudocode style. For clarity, main loops are commented. After the algorithm we included additional comments about some instructions that improve the search of frequent itemsets.

Algorithm bounded-neg-Apriori

1. set current floor $f := 0$
   set current level $\ell := 1$
   "This set is explained after the algorithm"
   $J := \emptyset$
2. "Initially, we find the frequent itemsets with isolated positive attributes"
   $L_{f,\ell} := \{A^{\{i\},\emptyset},\ \forall i \in I /\ fr(A^{\{i\},\emptyset}) > \sigma\}$
3. "This is the main loop to climb up floors"
   while $f \leq \ell$ and $\ell \leq n$ **do**
     while $L_{f,\ell} \neq \emptyset$ and $f \leq \ell$ and $\ell \leq n$ **do**
       $k := f + 1$
       $L_{\ell,\ell-1} := \emptyset$
       "At this moment we can obtain the frequent itemsets of the upper"
       "floors at same level from the itemsets in the previous floor"
       "There are two cases according to σ"
       while $k \leq \ell$ **do**
         if $k \notin J$ **then**
           $L_{k,\ell} := \emptyset$
           if $\sigma \leq 0.5$ **then** (1)
             $C_{k,\ell} := \{A^{p,s} / A^{p,s'} \in L_{k-1,\ell-1},\ m \in I - (p \cup s'),\ s = s' \cup \{m\},$
                 $\forall i \in p, A^{p-\{i\},s} \in L_{k,\ell-1},\ \forall j \in s, A^{p,s-\{j\}} \in L_{k-1,\ell-1}\}$
           else
             $C_{k,\ell} := \{A^{p,s} / A^{p,s'} \in L_{k-1,\ell-1},\ m \in I - (p \cup s'),\ s = s' \cup \{m\},$
                 $\forall i \in p, A^{p-\{i\},s} \in L_{k,\ell-1},\ \forall j \in s, A^{p,s-\{j\}} \in L_{k-1,\ell-1},$
                 $\forall j \in s, fr(A^{p\cup\{j\},s-\{j\}}) < \sigma\}$
           fi
           $L_{k,\ell} := \{A^{p,s} \in C_{k,\ell} / \exists j \in s,\ fr(A^{p,s-\{j\}}) > \sigma + fr(A^{p\cup\{j\},s-\{j\}})\}$
           if $L_{k,\ell} = \emptyset$ **then** $J := J \cup \{k\}$ **fi**
         fi
         if $\ell = 1$ and $k = 1$ and $L_{1,1} \neq \emptyset$ **then** $I := \{i / A^{\emptyset,\{i\}} \in L_{1,1}\}$ **fi** (2)
         set current floor $k := k + 1$

od (while k)
"Selected a floor we look for the frequent itemsets in next level"
"into this floor"
set current level $\ell := \ell + 1$
$J := J \cup \{k+1 / \, k \in J, k < n\}$
if $f = 0$ **then**
$C_{f,\ell} := \{A^{p,\emptyset} / \, \forall i \in p,\ A^{p-\{i\},\emptyset} \in L_{f,\ell-1}\}$
$L_{f,\ell} := \{A^{p,\emptyset} \in C_{f,\ell} / \, fr(A^{p,\emptyset}) > \sigma\}$
else
$C_{f,\ell} := \{A^{p,s} / \, \forall i \in p, A^{p-\{i\},s} \in L_{f,\ell-1},$
$\forall j \in s,\ A^{p,s-\{j\}} \in L_{f-1,\ell-1}\}$
$L_{f,\ell} := \{A^{p,s} \in C_{f,\ell} / \, fr(A^{p,s}) > \sigma\}$ (3)
fi
od (while ℓ)
"If the maximum level into a floor is reached then we must go over"
"to the next floor at this maximum level"
set current floor $f := f + 1$
$C_{f,\ell} := \{A^{p,s} / \, \forall i \in p, A^{p-\{i\},s} \in L_{f,\ell-1}, \forall j \in s,\ A^{p,s-\{j\}} \in L_{f-1,\ell-1}\}$
$L_{f,\ell} := \{A^{p,s} \in C_{f,\ell} / \, \exists j \in s, fr(A^{p,s-\{j\}}) > \sigma + fr(A^{p \cup \{j\},s-\{j\}})\}$
od (while f)
4. output $\bigcup_{k \le \ell \le n} L_{k,\ell}$

The algorithm refines the search of frequent itemsets by means of the set J. In each level, J indicates the floors where no frequent itemsets will exist.
In the sentence labeled (3) the generation of candidates and the computation of their frequencies must be done by considering σ (less or more than 0.5), as in the instruction labeled (1).

Note that, by the sentence labeled (2), the only negative attributes that could appear in the candidate itemsets are the elements of $L_{1,1}$. So, we use this set, as soon as it is computed, to refine the index set I used later along the computation.

With respect to the complexity of the algorithm, from a theoretical point of view, two aspects are considered: candidate generation and itemset frequence computation. In the candidate generation the worst case is reached when the threshold σ is less or equal to 0.5. In this case, two itemsets one of them with a particular attribute positive and the other itemset with the same attribute negative can be frequent simultaneously. If $\sigma > 0.5$ then by remark 2 in proposition 2 the generation is refined. Independtly of the σ value the sets I and J refine the candidate generation. So, the needed requirements can be reduced.

In the itemset frequence computation only itemsets with positive attributes are computed directly from the database. The frequencies of the other candidate itemsets with any number of negative attributes are obtained by using proposition one. Therefore, the number of passes through the database is like in Apriori, i.e., $n + 1$, where n is the greatest frequent itemset.

4 Conclusions and Future Work

In cases where the absence of some items from a transaction is relevant but one wants to avoid the generation of many rules relating these absences, it can be useful to allow for a maximum of k such absences from the frequent sets; even if no good guess exists for k, it may be useful to organize the search in such a way that the itemsets with m items show up in the order mandated by how many of them are positive: first all positive, then $m-1$ positive and one negative, and so on. Our algorithm allows one to do it and takes advantage of a number of facts, corresponding to relationships between the itemset frequencies, to avoid the counting of some candidates.

Of course, it makes sense to try to combine this strategy together with other ideas that have been used together with Apriori, like random sampling to evaluate the frequencies, or instead of Apriori, like alternative algorithms such as DIC [4] or Ready-and-Go [3]. Experimental developments, as well as more detailed analyses and a careful formalization of the setting, can lead to improved results, and we continue to work along these two lines.

References

1. Agrawal R., Imielinski T., Swami A.N.: Mining association rules between sets of items in large databases. *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD'93)*, ACM Press Washington D.C. , May 26-28 (1993) 207–216.
2. Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo A.I.: Fast discovery of association rules, in Fayyad U.M., Piatetsky-Shapiro G., Smyth Rp., Uthurusamy R. Eds, *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA; (1996) 307–328.
3. Baixeries J., Casas-Garriga G. and Balcázar J.L.: Frequent sets, sequences, and taxonomies: new, efficient algorithmic proposals. Tech. Rep. LSI-00-78-R. UPC. Barcelona (2000).
4. Brin S., Motwani R., Ullman J.D., Tsur S.: Dynamic Itemset Counting and Implication Rules for Market Basket Data. *Int. Conf. Management of Data*, ACM Press (1997) 255–264.
5. Fayyad U.M., Piatetsky–Shapiro G., Smyth P.: From data mining to knowledge discovery: An overview. In Fayyad U.M., Piatetsky–Shapiro G., Smyth P. and Uthurusamy R., eds, *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA, (1996) 1–34.
6. Gunopulos D., Khardon R., Mannila H., Toivonen H. Data Mining, Hypergraph Transversals, and Machine Learning. *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, ACM Press, Tucson, Arizona, May 12-14, (1997) 209–216.
7. Mannila H:, Toivonen H.: Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery.* **1(3)** (1997) 241–258.

Functional Trees

João Gama

LIACC, FEP - University of Porto
Rua Campo Alegre, 823
4150 Porto, Portugal
Phone: (+351) 226078830 Fax: (+351) 226003654
jgama@liacc.up.pt
http://www.niaad.liacc.up.pt/~jgama

Abstract. The design of algorithms that explore multiple representation languages and explore different search spaces has an intuitive appeal. In the context of classification problems, algorithms that generate multivariate trees are able to explore multiple representation languages by using decision tests based on a combination of attributes. The same applies to *model trees* algorithms, in regression domains, but using linear models at leaf nodes. In this paper we study *where* to use combinations of attributes in regression and classification tree learning. We present an algorithm for multivariate tree learning that combines a univariate decision tree with a linear function by means of constructive induction. This algorithm is able to use decision nodes with multivariate tests, and leaf nodes that make predictions using linear functions. Multivariate decision nodes are built when growing the tree, while functional leaves are built when pruning the tree. The algorithm has been implemented both for classification problems and regression problems. The experimental evaluation shows that our algorithm has clear advantages with respect to the generalization ability when compared against its components, two simplified versions, and competes well against the state-of-the-art in multivariate regression and classification trees.

Keywords: Decision Trees, Multiple Models, Supervised Machine Learning.

1 Introduction

The generalization ability of a learning algorithm depends on the appropriateness of its representation language to express a generalization of the examples for the given task. Different learning algorithms employ different representations, search heuristics, evaluation functions, and search spaces. It is now commonly accepted that each algorithm has its own selective superiority [3]; each is best for some but not all tasks. The design of algorithms that explore multiple representation languages and explore different search spaces has an intuitive appeal. This paper presents one such algorithm.

In the context of supervised learning problems it is useful to distinguish between classification problems and regression problems. In the former the target

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 59–73, 2001.

variable takes values in a finite and pre-defined set of un-ordered values, and the usual goal is to minimize a 0-1-loss function. In the later the target variable is ordered and takes values in a subset of $\Re$. The usual goal is to minimize a squared error loss function. Mainly due to the differences in the type of the target variable successful techniques in one type of problems are not directly applicable to the other type of problems.

The supervised learning problem is to find an approximation to an unknown function given a set of labelled examples. To solve this problem, several methods have been presented in the literature. Two of the most representative methods are the *General Linear Model* and *Decision trees*. Both methods explore different hypothesis space and use different search strategies. In the former the goal is to minimize the sum of squared deviations of the observed values for the dependent variable from those predicted by the model. It is based on the algebraic theory of invariants and has an analytical solution. The description language of the model takes the form of a polynomial that, in its simpler form, is a linear combination of the attributes: $w_0 + \sum w_i \times x_i$. This is the basic idea behind linear-regression and discriminant functions[8]. The latter use a *divide-and-conquer* strategy. The goal is to decompose a complex problem into simpler problems and recursively applying the same strategy to the sub-problems. Solutions of the sub-problems are combined in the form of a tree. Its hypothesis space is the set of all possible hyper-rectangular regions. The power of this approach comes from the ability to split the space of the attributes into subspaces, whereby each subspace is fitted with different functions. This is the basic idea behind well-known tree based algorithms [2,13].

In the case of classification problems, a class of algorithms that explore multiple representation languages are the so called *multivariate trees* [2,20,12,6,11]. In this sort of algorithms decision nodes can contain tests based on a combination of attributes. The language bias of univariate decision trees (axis parallel splits) are relaxed allowing decision surfaces oblique with respect to the axis of the instance space. As in the case of classification problems, in regression problems some authors have studied the use of regression trees that explore multiple representation languages, here denominated *model trees* [2,13,15,21,18]. But while in classification problems multivariate decisions appear in internal nodes, in regression problems multivariate decisions appear in leaf nodes. The problem that we study in this paper is *where* to use decisions based on combinations of attributes. Should we restrict combinations of attributes to decision nodes? Should we restrict combinations of attributes to leaf nodes? Could we use combinations of attributes both at decision nodes and leaf nodes?

The algorithm that we present here is an extension of multivariate trees. It is applicable to regression and classification domains, allowing combinations of attributes both at decision nodes and leaves. In the next section of the paper we describe our proposal to functional trees. In Section 3 we discuss the different variants of multivariate models using an illustrative example on regression domains. In Section 4 we present related work both in the classification and re-

gression settings. In Section 5 we evaluate our algorithm on a set of benchmark regression and classification problems. Last Section concludes the paper.

2 The Algorithm for Constructing Functional Trees

The standard algorithm to build univariate trees consists of two phases. In the first phase a large tree is constructed. In the second phase this tree is pruned back. The algorithm to grow the tree follows the standard divide-and-conquer approach. The most relevant aspects are: the splitting rule, the termination criterion, and the leaf assignment criterion. With respect to the last criterion, the usual rule consists of assignment of a constant to a leaf node. Considering only the examples that fall at this node, the constant is usually the majority class in classification problems or the mean of the y values in the regression setting. With respect to the splitting rule, each attribute value defines a possible partition of the dataset. We distinguish between nominal attributes and continuous ones. In the former the number of partitions is equal to the number of values of the attribute, in the latter a binary partition is obtained. To estimate the merit of the partition obtained by a given attribute we use the *gain ratio* heuristic for classification problems and the *decrease in variance* criterion for regression problems. In any case, the attribute that maximizes the criterion is chosen as test attribute at this node.

The pruning phase consists of traversing the tree in a depth-first fashion. At each non-leaf node two measures should be estimated. An estimate of the error of the subtree above this node, that is computed as a weighted sum of the estimated error for each leaf of the subtree, and the estimated error of the non-leaf node if it was pruned to a leaf. If the later is lower than the former, the entire subtree is replaced to a leaf.

All of these aspects have several and important variants, see for example [2, 14]. Nevertheless all decision nodes contain conditions based on the values of one attribute, and leaf nodes predict a constant.

2.1 Functional Trees

In this section we present the general algorithm to construct a functional tree. Given a set of examples and an attribute constructor, the main algorithm used to build a functional tree is presented in Figure 1. This algorithm is similar to many others, except in the constructive step (steps 2 and 3). Here a function is built and mapped to new attributes. There are some aspects of this algorithm that should be made explicit. In step 2, a model is built using the Constructor function. This is done using only the examples that fall at this node. Later, in step 3, the model is mapped to new attributes. Actually, the constructor function should be a classifier or a regressor depending on the type of the problem. In the case of regression problems the constructor function is mapped to one new attribute, the $\hat{y}$ value predict by the constructor. In the case of classification problems the number of new attributes is equal to the number of classes. Each

Function Tree(Dataset, Constructor)

1. If Stop_Criterion(DataSet)
   - Return a Leaf Node with a constant value.
2. Construct a model Φ using Constructor
3. For each example $\boldsymbol{x} \in DataSet$
   - Compute $\hat{y} = \Phi(\boldsymbol{x})$
   - Extend $\boldsymbol{x}$ with a new attribute $\hat{y}$.
4. Select the attribute from both original and all newly constructed attributes that maximizes some merit-function
5. For each partition i of the DataSet using the selected attribute
   - $\text{Tree}_i = \text{Tree}(\text{Dataset}_i, \text{Constructor})$
6. Return a *Tree*, as a decision node based on the select attribute, containing the Φ model, and descendents Tree_i.

End Function

Fig. 1. Building a Functional Tree

new attribute is the probability that the example belongs to one class[1] given by the constructed model. The merit of each new attribute is evaluated using the merit-function of the univariate tree, and in competition with the original attributes (step 4). The model built by our algorithm has two types of decision nodes: those based on a test of one of the original attributes, and those based on the values of the constructor function. When using Generalized Linear Models (GLM) [16] as attribute constructor, each new attribute is a linear combination of the original attributes. Decision nodes based on constructed attributes defines a multivariate decision surface.

Once a tree has been constructed, it is pruned back. The general algorithm to prune the tree is presented in Figure 2. To estimate the error at each leaf (step 1) we distinguish between classification and regression problems. In the former we assume a binomial distribution using a process similar to the *pessimistic error* of C4.5. In the latter we assume a χ^2 distribution of the variance of the cases in it using a process similar to the χ^2 pruning described in [18]. A similar procedure is used to estimate the constructor error (step 3). The pruning algorithm produces two different types of leaves: *Ordinary Leaves* that predict a constant, and *Constructor Leaves* that predict the value of the Constructor function learned (in the growing phase) at this node.

By simplifying our algorithm we obtain different conceptual models. Two interesting lesions are described in the following sub-sections.

Bottom-Up Approach. We denote as *Bottom-Up Approach* to functional trees when the functional models are used exclusively at leaves. This is the strategy

[1] At different nodes the system considers different number of classes depending on the class distribution of the examples that fall at this node.

Function Prune(Tree)

1. Estimate **Leaf_Error** as the error at this node.
2. If Tree is a leaf Return **Leaf_Error**.
3. Estimate **Constructor_Error** as the error of Φ [2].
4. For each descendent i
   - **Backed_Up_Error** += Prune(Tree$_i$)
5. If argmin(Leaf_Error,Constructor_Error,Backed_Up_Error)
   - Is Leaf_Error
     - Tree = Leaf
     - Tree_Error = Leaf_Error
   - Is Model_Error
     - Tree = Constructor Leaf
     - Tree_Error = Constructor_Error
   - Is Backed_Up_Error
     - Tree_Error = Backed_Up_Error
6. Return Tree_Error

End Function

Fig. 2. Pruning a Functional Tree

used for example in M5 [15,21], and in NBtree system [10]. In our tree algorithm this is done restricting the selection of the test attribute (step 4 in the growing algorithm) to the original attributes. Nevertheless we still build, at each node, the constructor function. The model built by the constructor function is used later in the pruning phase. In this way, all decision nodes are based in the original attributes. Leaf nodes could contain a constructor model. A leaf node contains a constructor model if and only if in the pruning algorithm the estimated error of the constructor model is lower than the *Backed-up-error* and the estimated error of the node has if a leaf replaced it.

Top-Down Approach. We denote as *Top-Down Approach* to functional trees when the multivariate models are used exclusively at decision nodes (internal nodes). In our algorithm, restricting the pruning algorithm to choose only between the **Backed_Up_Error** and the **Leaf_Error** obtain these kinds of models. In this case all leaves predict a constant value. This is the strategy used for example in systems like LMDT [20], OC1 [12], and Ltree [6].

Functional trees extend and generalize multivariate trees. Our algorithm can be seen as a hybrid model that performs a tight combination of a univariate tree and a GLM function. The components of the hybrid algorithm use different representation languages and search strategies. While the tree uses a divide-and-conquer method, the linear-regression performs a global minimization approach. While the former performs feature selection, the later uses all (or almost all) the attributes to build a model. From the point of view of the bias-variance

decomposition of the error [1] a decision tree is known to have low bias but high variance, while GLM functions are known to have low variance but high bias. This is the desirable behaviour for components of hybrid models.

3 An Illustrative Example

In this section we use the well-known regression dataset *Housing* to illustrate the different variants of functional models. The attribute constructor used is the linear regression function. Figure 3(a) presents a univariate tree for the *Housing*

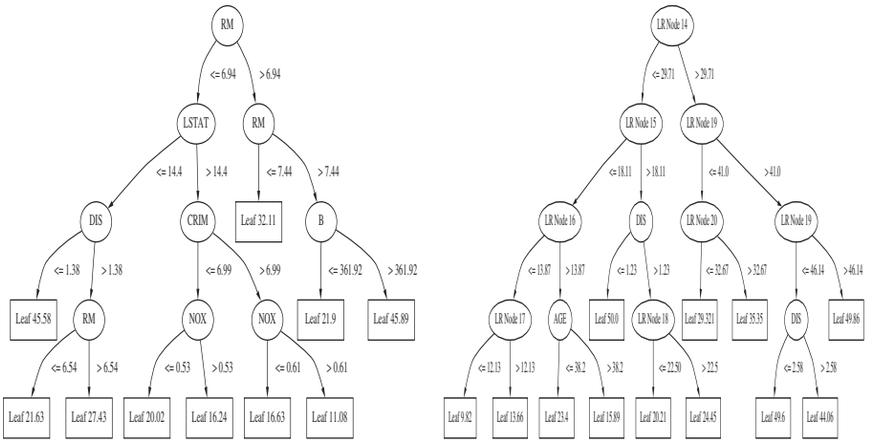


Fig. 3. (a)The Univariate Regression Tree and (b) Top-Down regression tree for the Housing problem.

dataset. Decision nodes only contain tests based on the original attributes. Leaf nodes predict the average of y values taken from the examples that fall at the leaf.

In a top-down multivariate tree (Figure 3(b)) decision nodes could contain (not necessarily) tests based on a linear combination of the original attributes. The tree contains a mixture of learned attributes, denoted as *LR Node*, and original attributes, *e.g. AGE, DIS*. Any of the linear-regression attributes can be used both at the node where they have been created and at deeper nodes. For example, the *LR Node 19* has been created at the second level of the tree. It is used as test attribute at this node, and also (due to the constructive ability) as test attribute at the third level of the tree. Leaf nodes predict the average of y values of the examples that fall at this leaf. In a bottom-up multivariate tree (Figure 4(a)) decision nodes only contain tests based on the original attributes. Leaf nodes could predict (not necessarily) values obtained by using a linear-regression function built from the examples that fall at this node. This is

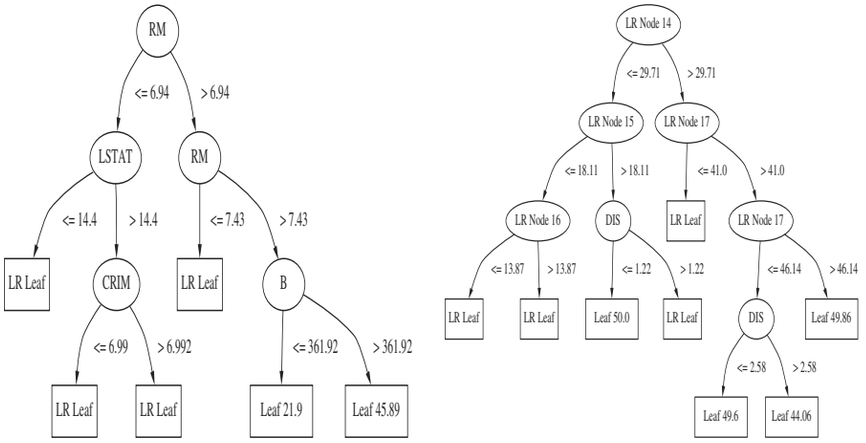


Fig. 4. (a)The Bottom-Up Multivariate Regression Tree and (b) The Multivariate Regression Tree for the Housing problem.

the kind of multivariate regression trees that usually appears on the literature. For example, systems M5 [15,21] and RT [18] generate this kind of models. Figure 4(b) presents the full multivariate regression tree using both top-down and bottom-up multivariate approaches. In this case, decision nodes could contain (not necessarily) tests based on a linear combination of the original attributes, and leaf nodes could predict (not necessarily) values obtained by using a linear-regression function built from the examples that fall at this node.

Figure 5 illustrates the functional models in the case of a classification problem. We have used the UCI dataset *Learning Qualitative Structure Activity Relationships - QSARs pyrimidines* to illustrate the different variants of tree models. This is a complex two classes problem defined by 54 continuous attributes. The attribute constructor used is the LinearBayes [5] classifier. In a bottom-up functional tree (Figure 5(a)) decision nodes only contain tests based on the original attributes. Leaf nodes could predict (not necessarily) values obtained by using a LinearBayes function built from the examples that fall at this node. Figure 5(b) presents the functional tree using both top-down and bottom-up multivariate approaches. In this case, decision nodes could contain (not necessarily) tests based on a linear combination of the original attributes, and leaf nodes could predict (not necessarily) values obtained by using a LinearBayes function built from the examples that fall at this node.

4 Related Work

Breiman *et.al.* [2] presents the first extensive and in-depth study of the problem of constructing decision and regression trees. But, while in the case of decision trees they consider internal nodes with a test based on linear combination of

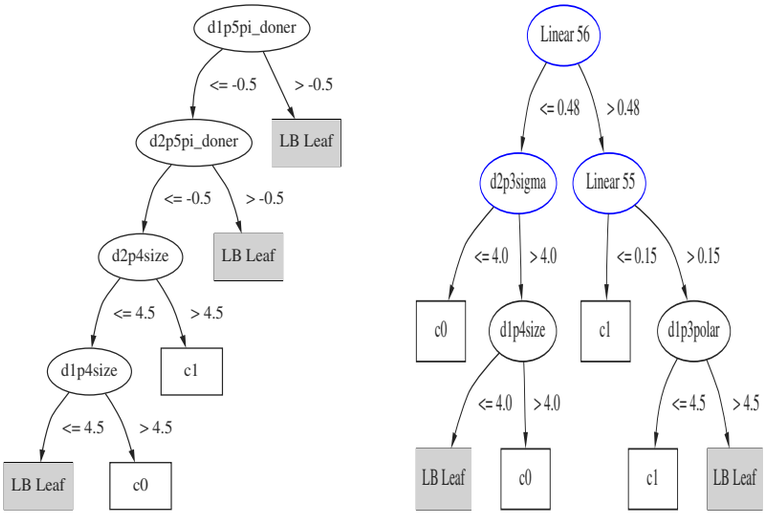


Fig. 5. (a)The Bottom-Up Functional Tree and (b) the Functional Tree for the QSARs problem.

attributes, in the case of regression trees internal nodes are always based on a single attribute.

In the context of classification problems, several algorithms have been presented that could use at each decision node tests based on linear combination of the attributes [2,12,20,6]. The most comprehensive study on multivariate trees has been presented by Brodley and Utgoff in [4]. Brodley and Utgoff discusses several methods for constructing multivariate decision trees: representing a multivariate test, including symbolic and numeric features, learning the coefficients of a multivariate test, selecting the features to include in a test, and pruning of multivariate decision trees. Brodley only considers multivariate tests at inner nodes in a tree. In this context few works consider functional tree leaves. One of the earliest work is the Percepton tree algorithm [19] where leaf nodes may implement a general linear discriminant function. Also Kohavi[10] has presented the naive Bayes tree that uses functional leaves. NBtree is a hybrid algorithm that generates a regular univariate decision tree, but the leaves contain a naive Bayes classifier built from the examples that fall at this node. The approach retains the interpretability of naive Bayes and decision trees, while resulting in classifiers that frequently outperform both constituents, especially in large datasets. Also, Gama [7] has presented *Cascade Generalization*, a method to combine classification algorithms by means of constructive induction. The work presented here, near follows Cascade method but extended for regression domains and allowing models with functional leaves.

In regression domains, Quinlan [13] has presented system M5. It builds multivariate trees using linear models at the leaves. In the pruning phase for each

leaf a linear model is built. Recently, Witten and Eibe [21] have extended M5. A linear model is built at each node of the initial regression tree. All the models along a particular path from the root to a leaf node are then combined into one linear model in a *smoothing* step. Also Karalic [9] has studied the influence of using linear regression in the leaves of a regression tree. As in the work of Quinlan, Karalic shows that it leads to smaller models with increase of performance. Torgo [17] has presented an experimental study about functional models for regression tree leaves. Later, the same author [18] has presented the system RT. Using RT with linear models at the leaves, RT builds and prunes a regular univariate tree. Then at each leaf a linear model is built using the examples that fall at this leaf.

5 Experimental Evaluation

It is commonly accepted that multivariate regression trees should be competitive against univariate models. In this section we evaluate the proposed algorithm, its simplified variants, and its components on a set of classification and regression benchmark problems. In regression problems the constructor is a standard linear regression function. In classification problems the constructor is the LinearBayes classifier [5]. For comparative proposes we evaluate also system M5[3]. The main goal in this experimental evaluation is to study the influence in terms of performance of the position inside a regression and a classification tree of the linear models. We evaluate three situations:

- Trees that could use linear combinations at each internal node.
- Trees that could use linear combinations at each leaf.
- Trees that could use linear combinations both at each internal and leaf nodes.

All evaluated models are based on the same tree growing and pruning algorithm. That is, they use exactly the same splitting criteria, stopping criteria, and pruning mechanism. Moreover they share many minor heuristics that individually are too small to mention, but collectively can make difference. Doing so, the differences on the evaluation statistics are due to the differences in the conceptual model.

In this work we estimate the performance of a learned model using 10 fold cross validation. To minimize the influence of the variability of the training set, we repeat this process ten times, each time using a different permutation of the dataset. The final estimate is the mean of the performance statistic obtained in each run of the cross validation. For regression problems the performance is measured in terms of the *mean squared error* statistic. For classification problems the performance is measured in terms of the *error rate* statistic. To apply pairwise comparisons we guarantee that, in all runs, all algorithms learn and test on the same partitions of the data. We compare the performance of the

[3] We have used M5 from version 3.1.8 of the Weka environment. We have used several regression systems. The most competitive was M5.

functional tree (FT) against its components: the univariate tree (UT) and the constructor function (linear regression (LR) in regression problems, and LinearBayes (LB) in classification problems). The functional tree is also compared against to the two simplified versions: Bottom-up (FT-B) and Top-Down (FT-T). For each dataset, comparisons between algorithms are done using the *Wilcoxon signed ranked paired-test*. The null hypothesis is that the difference between performance statistics has median value zero. We consider that a difference in performance has statistical significance if the *p value* of the Wilcoxon test is less than 0.01.

5.1 Results in Regression Domains

We have chosen 20 datasets from the *Repository of Regression problems at LIACC* [4]. The choice of datasets was restricted by the criteria that almost all the attributes are ordered with few missing values[5]. The number of examples varies from 43 to 40768. The number of attributes varies from 5 to 48. The results in terms of MSE and standard deviation are presented in Table 1. The first two columns refer to the results of the components of the hybrid algorithm. The following three columns refer to the simplified versions of our algorithm and the full model. The last column refers to the M5 system. For each dataset, the algorithms are compared against the full multivariate tree using the *Wilcoxon signed rank-test*. A $-$ $(+)$ sign indicates that for this dataset the performance of the algorithm was worse (better) than the full model with a *p value* less than 0.01.

Table 1 presents a comparative summary of the results. The first line presents the geometric mean of the MSE statistic across all datasets. The second line shows the average rank of all models, computed for each dataset by assigning rank 1 to the best algorithm, 2 to the second best and so on. The third line shows the average ratio of *MSE*. This is computed for each dataset as the ratio between the *MSE* of one algorithm and the *MSE* of M5. The fourth line shows the number of significant differences using the *signed-rank test* taking the multivariate tree FT as reference. We use the *Wilcoxon Matched-Pairs Signed-Ranks Test* to compare the error rate of pairs of algorithms across datasets[6]. The last line shows the *p values* associated with this test for the *MSE* results on all datasets and taking FT as reference. It is interesting to note that the full model (FT) significantly improves over both components (LR and UT) in 14 datasets out of 20. All the multivariate trees have a similar performance. Using the significant test as criteria, FT is the most performing algorithm. It is interesting to note that the bottom-up version is the most competitive algorithm. The ratio of significant wins/losses between the bottom-up and top-down versions is 4/3.

[4] http://www.ncc.up.pt/∼ltorgo/Datasets

[5] In regression problems, the actual implementation ignores missing values at learning time. At application time, if the value of the test attribute is unknown, all descendent branches produce a prediction. The final prediction is a weighted average of the predictions.

[6] Each pair of data points consists of the estimate MSE on one dataset and for the two learning algorithms being compared.

Table 1. Summary of Results in Regression Problems (MSE).

	L.Regression	Univ. Tree	Functional Trees			
Data	(LR)	(UT)	**Top**	**Bottom**	**FT**	**M5**
Abalone	− 4.908±0.0	− 5.728±0.1	4.616±0.0	− 4.759±0.0	4.602±0.0	4.553±0.5
Auto-mpg	− 11.470±0.1	− 19.409±1.2	+ 8.921±0.4	9.560±0.8	9.131±0.5	7.958±3.5
Cart	− 5.684±0.0	+ 0.995±0.0	− 1.016±0.0	+ 0.993±0.0	1.012±0.0	0.994±0.0
Computer	− 99.907±0.2	− 10.955±0.6	− 6.426±0.6	− 6.507±0.5	6.284±0.6	− 8.081±2.7
Cpu	− 3734±1717	− 4111±1657	− 1760±389	− 1197±161	1070±137	1092±1315
Diabetes	0.399±0.0	− 0.535±0.0	− 0.500±0.0	0.400±0.0	0.399±0.0	0.446±0.3
Elevators	− 1.02e-5±0.0	− 1.4e-5±0.0	− 0.86e-5±0.0	0.5e-5±0.0	0.5e-5±0.0	0.52e-5±0.0
Fried	− 6.924±0.0	− 3.474±0.0	− 1.862±0.0	− 2.348±0.0	1.850±0.0	− 1.938±0.1
H.Quake	0.036±0.0	0.036±0.0	0.036±0.0	0.036±0.0	0.036±0.0	0.036±0.0
House(16H)	−2.06e9±6.1e5	− 1.69e9±3.3e7	+ 1.20e9±2.2e7	1.19e9±3.0e7	1.23e9±2.2e7	1.27e9±1.2e8
House(8L)	−1.73e9±8.2e5	− 1.19e9±1.2e7	+ 1.01e9±1.3e7	1.02e9±9.2e6	1.02e9±1.3e7	9.97e8±7.1e7
House(Cal)	−4.81e9±2.0e6	− 3.69e9±3.5e7	− 3.09e9±2.7e7	+ 2.78e9±2.8e7	3.05e9±3.1e7	3.07e9±2.8e8
Housing	− 23.840±0.2	− 19.591±1.7	16.251±1.1	+ 13.359±1.7	16.538±1.3	12.467±7.5
Kinematics	− 0.041±0.0	− 0.035±0.0	− 0.027±0.0	− 0.026±0.0	0.023±0.0	− 0.025±0.0
Machine	− 5952±2053	− 6036±1752	3473±673	3300±757	3032±759	3557±4271
Pole	− 930.08±0.3	+ 48.55±1.2	79.48±2.6	+ 35.16±0.7	79.31±2.4	+ 42.0±5.8
Puma32	− 7.2e-4±0.0	− 1.1e-4±0.0	+ 0.71e-4±0.0	0.82e-4±0.0	0.82e-4±0.0	0.67e-4±0.0
Puma8	− 19.925±0.0	− 13.307±0.2	+ 11.047±0.1	11.145±0.1	11.241±0.1	+ 10.299±0.5
Pyrimidines	− 0.018±0.0	0.014±0.0	+ 0.010±0.0	0.013±0.0	0.013±0.0	0.012±0.0
Triazines	− 0.025±0.0	+ 0.019±0.0	− 0.018±0.0	0.023±0.0	0.023±0.0	0.017±0.0

Summary of MSE Results

	LR	**UT**	**FT-T**	**FT-B**	**FT**	**M5**
Geometric Mean	39.2	23.59	17.68	16.47	16.90	16.2
Average Rank	5.4	4.9	3.15	2.9	2.5	2.3
Average Ratio	4.0	1.57	1.13	1.03	1.07	1
Wins / Losses	1/19	4/16	8/12	6/11	–	11/9
Signi. Wins/Losses	0/18	3/15	6/9	4/5	–	2/3
Wilcoxon Test	0.0	0.02	0.21	0.1	–	0.23

Nevertheless there is a computational cost associated with the increase in performance verified. To run all the experiments referred here, FT requires almost 1.8 more time than the univariate regression tree.

5.2 Results in Classification Problems

We have chosen 30 datasets from the UCI repository. For comparative purposes we also evaluate M5′ [21]. M5′ decomposes a n-classes classification problem into $n-1$ binary regression problems[7]. The results in terms of error-rate and standard deviation are presented in Table 2. The first two columns refer to the results of the components of our system, the LinearBayes and the univariate tree. The next two columns refer to the lesioned versions of the algorithm, the Bottom-Up (FT-B) and Top-Down (FT-T). The fifth column refers to the full proposed

[7] We have used other multivariate trees. The most competitive was M5′.

Table 2. Summary of Error Rate Results

	LinBayes	**Univ. Tree**	**Functional Trees**			
Dataset	**LB**	**UT**	**Bottom**	**Top**	**FT**	**M5′**
Adult	− 17.012±0.5	14.178±0.5	− 14.307±0.4	13.800±0.4	13.830±0.4	− 15.182±0.6
Australian	13.498±0.3	14.750±1.0	− 14.343±0.4	13.928±0.6	13.638±0.6	14.643±5.2
Balance	− 13.355±0.3	− 22.467±1.1	− 10.445±0.6	7.313±0.9	7.313±0.9	− 13.894±3.2
Banding	23.681±1.0	23.512±1.8	23.512±1.8	23.762±2.2	23.762±2.2	22.619±5.3
Breast(W)	+ 2.862±0.1	− 5.123±0.2	− 4.337±0.1	3.346±0.4	3.346±0.4	5.137±3.1
Cleveland	16.134±0.4	− 20.995±1.4	+ 15.952±0.5	17.369±0.9	16.675±0.8	17.926±8.0
Credit	+ 14.228±0.1	14.608±0.5	14.784±0.5	15.103±0.4	15.220±0.6	14.913±3.7
Diabetes	+ 22.709±0.2	− 25.348±1.0	23.998±1.0	− 25.206±0.9	23.658±1.0	25.002±4.8
German	24.520±0.2	28.240±0.7	+ 23.630±0.5	24.870±0.5	24.330±0.7	26.300±3.1
Glass	− 36.647±0.8	32.150±2.3	32.150±2.3	32.509±3.3	32.509±3.3	29.479±10.4
Heart	17.704±0.2	− 23.074±1.7	17.037±0.6	17.333±1.4	17.185±0.8	16.667±8.9
Hepatitis	+ 15.481±0.7	17.135±1.3	17.135±1.3	17.135±1.3	17.135±1.3	19.919±8.5
Ionosphere	13.379±0.8	10.025±0.9	10.624±0.9	11.175±1.4	11.175±1.4	9.704±4.1
Iris	2.000±0.0	− 4.333±0.8	2.067±0.2	− 3.733±0.8	2.067±0.2	5.333±5.3
Letter	− 29.821±1.3	11.880±0.6	12.005±0.6	11.799±1.1	11.799±1.1	+ 9.440±0.5
Monks-1	− 25.009±0.0	10.536±1.7	11.150±1.9	8.752±1.9	8.729±1.9	10.054±8.9
Monks-2	− 34.186±0.6	− 32.865±0.0	− 33.907±0.4	9.004±1.6	9.074±1.6	27.664±20.9
Monks-3	− 4.163±0.0	+ 1.572±0.4	3.511±0.9	2.884±0.4	2.998±0.4	1.364±2.4
Mushroom	− 3.109±0.0	+ 0.000±0.0	+ 0.062±0.0	0.112±0.0	0.112±0.0	0.025±0.1
Optdigits	− 4.687±0.1	− 9.476±0.3	− 4.732±0.1	3.295±0.1	3.300±0.1	− 5.429±1.4
Pendigits	− 12.425±0.0	− 3.559±0.1	− 3.099±0.1	2.890±0.1	2.890±0.1	2.419±0.4
Pyrimidines	− 9.846±0.1	+ 5.733±0.2	6.115±0.2	6.158±0.2	6.159±0.2	6.175±0.9
Satimage	− 16.011±0.1	− 12.894±0.2	− 12.894±0.2	11.776±0.3	11.776±0.3	12.402±3.2
Segment	− 8.407±0.1	3.381±0.2	3.381±0.2	3.190±0.2	3.190±0.2	2.468±0.8
Shuttle	− 5.629±0.3	0.028±0.0	0.028±0.0	0.036±0.0	0.036±0.0	0.067±0.0
Sonar	24.955±1.2	27.654±3.5	27.654±3.5	27.654±3.5	27.654±3.5	22.721±9.0
Vehicle	22.163±0.1	− 27.334±1.2	+ 18.282±0.5	21.090±1.1	21.031±1.1	20.900±4.6
Votes	− 9.739±0.2	3.773±0.5	3.773±0.5	3.795±0.5	3.795±0.5	4.172±4.0
Waveform	+ 14.939±0.2	− 24.036±0.8	+ 15.216±0.2	− 16.142±0.3	15.863±0.4	− 17.241±1.4
Wine	1.133±0.5	− 6.609±1.3	1.404±0.3	1.459±0.3	1.404±0.3	3.830±3.6

	LB	**UT**	**FT-B**	**FT-T**	**FT**	**M5′**
Average Mean	15.31	14.58	12.72	11.89	11.72	12.77
Geometric Mean	11.63	9.03	7.03	6.80	6.63	7.24
Average Rank	4.0	4.1	3.1	3.3	3.0	3.4
Average Ratio	7.545	1.41	1.12	1.032	1	1.23
Wins/Losses	11/19	9/19	13/13	6/10	–	12/18
Significant Wins/Losses	5/15	3/12	5/8	0/3	–	1/4
Wilcoxon Test	0.00	0.00	0.8	0.07	–	

model(FT). The last column refers to the results of M5′. For each dataset, the algorithms are compared against the full functional tree using the *Wilcoxon signed rank-test*. A $-$ (+) sign indicates that for this dataset the performance of the algorithm was worse (better) than the full model with a *p value* less than 0.01.

Table 2 present a comparative summary of the results. The first two lines present the arithmetic and the geometric mean of the error rate across all datasets. The third line shows the average rank of all models, computed for each dataset by assigning rank 1 to the best algorithm, 2 to the second best and so on. The fourth line shows the average ratio of error rates. This is computed for each dataset as the ratio between the error rate of one algorithm and the error rate of the full functional tree FT. The fifth line shows the number of significant differences using the *signed-rank test* taking the multivariate tree FT as reference. We use the *Wilcoxon Matched-Pairs Signed-Ranks Test* to compare the error rate of pairs of algorithms across datasets. The last line shows the *p values* associated with this test for the results on all datasets and taking FT as reference. All the evaluation statistics shows that FT is a competitive algorithm. The most competitive simplified version is, again, the bottom-up version. The ratio of significant wins/losses between the bottom-up and top-down versions is 10/6. It is interesting to note that the full model (FT) significantly improves over both components (LB and UT) in 6 datasets.

5.3 Discussion

The experimental evaluation points out some interesting observations:

- For both types of problems we obtain similar rankings of the performance between the different versions of the algorithms.
- All multivariate trees versions have similar performance. On these datasets, there is no clear winner between the different versions of functional trees.
- Any functional tree out-performs its constituents in a large set of problems.

In our study the results are consistent on both type of problems. Our experimental study suggests that the full model, that is a multivariate model using linear functions *both* at decision nodes and leaves, is the most performing algorithm. Another dimension of analysis is the size of the model. Here we consider the number of leaves. This measures the number of different regions into which the instance space is partitioned. On this datasets, the average number of leaves for the univariate tree is 70. Any multivariate tree generates smaller models. The average number of leaves of the full model is 50, for the bottom approach is 56, and for the top approach is 52. Nevertheless there is a computational cost associated with the increase in performance verified. To run all the experiments referred here, FT requires almost 1.7 more time than the univariate tree.

6 Conclusions

In this paper we have presented Functional Trees, a new formalism to construct multivariate trees for regression and classification problems. The proposed algo-

rithm is able to use functional decision nodes and functional leaf nodes. Functional decision nodes are built when growing the tree, while functional leaves are built when pruning the tree. A contribution of this work is that it provides a single framework for classification and regression multivariate trees. Functional trees can be seen as a generalization of multivariate trees for decision problems and model-trees for regression problems, allowing functional decisions both at inner and leaf nodes. We have experimentally observed that the unified framework is competitive against the state-of-the-art in model-trees.

Another contribution of this work is the study about *where* to use decisions based on a combination of attributes both in regression and classification. In the experimental evaluation on a set of benchmark problems we have compared the performance of a functional tree against its components, two simplified versions and the state-of-the-art in multivariate trees. The results are consistent on both type of problems. Our experimental study suggests that the full model, that is a multivariate model using linear functions *both* at decision nodes and leaves, is the most performing algorithm. Although most of the work in multivariate classification trees follows the top-down approach, the bottom-up approach seems to be competitive. A similar observation applies to regression problems. This observation point directions for future research on this topic.

Acknowledgments. Gratitude is expressed to the financial support given by the FEDER and PRAXIS XXI, the Plurianual support attributed to LIACC, and Esprit LTR METAL project, the project Data Mining and Decision Support for Business Competitiveness (Sol-Eu-Net), and project ALES. I would like to thank the anonymous reviewers for the constructive comments.

References

1. L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–849, 1998.
2. L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group., 1984.
3. Carla E. Brodley. Recursive automatic bias selection for classifier construction. *Machine Learning*, 20:63–94, 1995.
4. Carla E. Brodley and Paul E. Utgoff. Multivariate decision trees. *Machine Learning*, 19:45–77, 1995.
5. J. Gama. A Linear-Bayes classifier. In C. Monard, editor, *Advances on Artificial Intelligence -SBIA2000*. LNAI 1952 Springer Verlag, 2000.
6. João Gama. Probabilistic Linear Tree. In D. Fisher, editor, *Machine Learning, Proceedings of the 14th International Conference*. Morgan Kaufmann, 1997.
7. João Gama and P. Brazdil. Cascade Generalization. *Machine Learning*, 41:315–343, 2000.
8. Geoffrey J.Mclachlan. *Discriminant Analysis and Statistical Pattern Recognition*. New York, Willey and Sons, 1992.
9. Aram Karalic. Employing linear regression in regression tree leaves. In Bernard Neumann, editor, *European Conference on Artificial Intelligence*, 1992.

10. R. Kohavi. Scaling up the accuracy of naive Bayes classifiers: a decision tree hybrid. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996.
11. W. Loh and Y. Shih. Split selection methods for classification trees. *Statistica Sinica*, 7:815–840, 1997.
12. S. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 1994.
13. R. Quinlan. Learning with continuous classes. In Adams and Sterling, editors, *Proceedings of AI'92*. World Scientific, 1992.
14. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc., 1993.
15. R. Quinlan. Combining instance-based and model-based learning. In P.Utgoff, editor, *ML93, Machine Learning, Proceedings of the 10th International Conference*. Morgan Kaufmann, 1993.
16. Paul Taylor. Statistical methods. In M. Berthold and D. Hand, editors, *Intelligent Data Analysis - An Introduction*. Springer Verlag, 1999.
17. Luis Torgo. Functional models for regression tree leaves. In D. Fisher, editor, *Machine Learning, Proceedings of the 14th International Conference*. Morgan Kaufmann, 1997.
18. Luis Torgo. *Inductive Learning of Tree-based Regression Models*. PhD thesis, University of Porto, 2000.
19. P. Utgoff. Percepton trees - a case study in hybrid concept representation. In *Proceedings of the Seventh National Conference on Artificial Intelligence*. Morgan Kaufmann, 1988.
20. P. Utgoff and C. Brodley. Linear machine decision trees. Coins technical report, 91-10, University of Massachusetts, 1991.
21. Ian Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers, 2000.

Spherical Horses and Shared Toothbrushes: Lessons Learned from a Workshop on Scientific and Technological Thinking

Michael E. Gorman[1], Alexandra Kincannon[2], and Matthew M. Mehalik[1]

[1]Technology, Culture & Communications, School of Engineering & Applied Science, P.O. Box 400744, University of Virginia, Charlottesville, VA 22904-4744 USA
{meg3c, mmm2f}@virginia.edu
[2]Department of Psychology, P.O. Box 400400, University of Virginia, Charlottesville, VA 22904-4400 USA
kincannon@virginia.edu

Abstract. We briefly summarize some of the lessons learned in a workshop on cognitive studies of science and technology. Our purpose was to assemble a diverse group of practitioners to discuss the latest research, identify the stumbling blocks to advancement in this field, and brainstorm about directions for the future. Two questions became central themes. First, how can we combine artificial studies involving 'spherical horses' with fine-grained case studies of actual practice? Results obtained in the laboratory may have low applicability to real world situations. Second, how can we deal with academics' attachments to their theoretical frameworks? Academics often like to develop unique 'toothbrushes' and are reluctant to use anyone else's. The workshop illustrated that toothbrushes can be shared and that spherical horses and fine-grained case studies can complement one another. Theories need to deal rigorously with the distributed character of scientific and technological problem solving. We hope this workshop will suggest directions more sophisticated theories might take.

1 Introduction

At the turn of the 21st century, the most valuable commodity in society is knowledge, particularly new knowledge that may give a culture, a company, or a laboratory an advantage [1-3]. Therefore, it is vital for the science and technology studies community to study the thinking processes that lead to discovery, new knowledge and invention. Knowledge about these processes can enhance the probability of new and useful technologies, clarify the process by which new ideas are turned into marketable realities, make it possible for us to turn students into ethical inventors and entrepreneurs, and facilitate the development of business strategies and social policies based on a genuine understanding of the creative process.

2 A Workshop on Scientific and Technological Thinking

In order to get access to cutting-edge research on techno-scientific thinking, Michael Gorman obtained funding from the National Science Foundation, the Strategic Institute of the Boston Consulting Group and the National Collegiate Inventors and Innovators Alliance to hold a workshop at the University of Virginia from March 24-27, 2001. With assistance from Alexandra Kincannon, Ryan Tweney and others, he as

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 74-86, 2001.

sembled a diverse group of practitioners, focusing on those in the middle of their careers and also on junior faculty and graduate students who represent the future. There were 29 participants, including 18 senior or mid-career researchers, and 11 junior faculty and graduate students. Representatives from the NSF, the Strategic Institute of the BCG and the NCIIA also attended. Their role was to keep participants focused on lessons learned, even as the participants worked to assess the state of the art and push beyond it, establishing new directions for research on scientific and technological thinking.

In the rest of this brief paper, Gorman and Kincannon, two of the organizers of the workshop, and Matthew Mehalik, one of the participants, will highlight results from this workshop, citing the work of participants where appropriate and adding interpretive material of their own.[1]

Two questions dominated in the workshop, each illustrated by a metaphor. David Gooding, a philosopher from the University of Bath who has done fine-grained studies of the thinking processes of Michael Faraday, told a joke that set up one theme. In the joke, a multimillionaire offered a prize for predicting the outcome of a horse race to a stockbreeder, a geneticist, and a physicist. The stockbreeder said there were too many variables, the geneticist could not make a prediction about any particular horse, but the physicist claimed the prize, saying he could make the prediction to many decimal places—provided it were a perfectly spherical horse moving through a vacuum. This metaphor led to a question: How can we combine artificial studies involving 'spherical horses' and fine-grained case studies of actual practice? Results obtained under rigorous laboratory conditions may have what psychologists call low ecological validity, or low applicability to real world situations [4]. Highly abstract computational models often ignore the way in which real-world knowledge is embedded in social contexts and embodied in hands-on practices [5].

The second metaphor came from Christian Schunn, then at George Mason University and now at the University of Pittsburgh, who noted that taxonomies and frameworks are like toothbrushes—no one wants to use anyone else's. This metaphor led to another question: How can we transcend academics' attachments to their individual theoretical frameworks? Academic psychologists, historians, sociologists and philosophers like to develop and refine unique toothbrushes and are reluctant to use anyone else's. Real-world practitioners are not as fussy; they are willing to assemble a 'bricolage' of elements from various frameworks that academics might regard as incommensurable.

3 A Moratorium against Spherical Horses?

Nancy Nersessian, a philosopher and cognitive scientist from the Georgia Institute of Technology, reminded participants that Bruno Latour declared a ten-year moratorium against cognitive studies of science in 1986. Latour was one of the key figures in promoting a new sociology of scientific knowledge. He and others were reacting

[1] The views reflected here are those of the authors, and have not been endorsed by workshop participants, the NSF, BCG or the NCIIA. All participants were taped, with their consent, and we have used these tapes in an effort to reconstruct highlights. Thanks to Pat Langley for his comments on a draft.

against the idea that science was a purely rational enterprise, carried out in an abstract cognitive space.

Cognitive scientists like Herbert Simon contributed to this abstract cognizer view of science.[2] Simon was one of the founders of a movement Nersessian labeled "Good Old Fashioned Artificial Intelligence" (GOFAI). Simon's toothbrush, or framework, began with the assumption that there is nothing particularly unique about what a Kepler does—the same thinking processes are used on both ordinary and extraordinary problems [6]. Simon was a revolutionary in the Kuhnian sense; he played a major role in creating artificial intelligence and linking it with a new science of thinking, called cognitive science.

Peter Slezak used programs like BACON to turn the tables on Latour's moratorium:

> A decisive and sufficient refutation of the 'strong programme' in the sociology of scientific knowledge (SSK) would be the demonstration of a case in which scientific discovery is totally isolated from all social or cultural factors whatever. I want to discuss examples where precisely this circumstance prevails concerning the discovery of fundamental laws of the first importance in science. The work I will describe involves computer programs being developed in the burgeoning interdisciplinary field of cognitive science, and specifically within 'artificial intelligence' (AI). The claim I wish to advance is that these programs constitute a 'pure' or socially uncontaminated instance of inductive inference, and are capable of autonomously deriving classical scientific laws from the raw observational data [7, pp. 563-564].

Slezak argued that if programs like BACON [8, 9] can discover, then there is no need to invoke all these interests and negotiations the sociologists use to explain discovery. His claims sparked a vigorous debate in the November, 1989 issue of the journal *Social Studies of Science*. Latour and Slezak illustrate how academics can create almost incommensurable frameworks. If the Simon perspective is a toothbrush, then Latour is denying that it even exists—and vice versa.

Nersessian reminded participants that, had Simon been at the workshop, he would have argued that his toothbrush does incorporate the social and cultural; it is just that all of this is represented symbolically in memory [10]. Therefore, cognition is about symbol processing. These symbols could be as easily instantiated in a computer as in a brain.

In contrast, Greeno and others advocate a position whose roots might be traced to Gibson and Dewey: that knowledge emerges from the interaction between the individual and the situation [11]. Cognition is distributed in the environment as well as the brain, and is shared among individuals [12, 13]. Merlin Donald discusses the role of culture in the evolution of cognition [14]. Nersessian in her own work explored how cultural factors can account for differences between the problem-solving approaches of scientists like Maxwell and Ampere [15].

In the symbol-processing view, discovery and invention are merely aspects of a general problem-solving system that can best be represented at the 'spherical horse' level. In the situated and distributed view, discovery and invention are practices that

---

[2] Simon intended to be a participant in our workshop, but died shortly before it—a great tragedy and a great loss. During the planning stages, he referred to this as a workshop of 'right thinkers'. For tributes to him, see http://www.people.virginia.edu/~apk5t/STweb/mainST.html.

need to be studied in their social context. This situated cognition perspective comes much closer to that of sociologists and anthropologists of science [16], but advocates like Norman and Hutchins still talk about the importance of representations like mental models.

Jim Davies, from the Georgia Institute of Technology, applied Nersessian's cognitive-historical approach to a case study of the use of visual analogy in scientific discovery. Davies analyzed the process of conceptual change in Maxwell's work on electromagnetism and applied to it a model of visual analogical problem solving called Galatea. He found that visual analogy played an important role in the development of Maxwell's theories and demonstrated that the cognitive-historical approach is useful for understanding general cognitive processes.

Ryan Tweney, a co-organizer of the workshop, described his own in vivo case study of Michael Faraday's work on the interaction of light and gold films [33]. Tweney is in the process of replicating these experiments to unpack the tacit knowledge that is embodied in the cognitive artifacts created by Faraday. He hopes to do a kind of material protocol analysis that goes beyond the verbal material that is in Faraday's diary. One end result might be a digital version of Faraday's diary that includes images and perhaps even QuickTime movies of replications of his experiments. This kind of study potentially bridges the gap between situated and symbolic studies of discovery.

4 A Common Set of Toothbrushes?

David Klahr, a cognitive psychologist at Carnegie Mellon, has shown a preference for spherical horses, conducting experiments on scientific thinking. However, his experiments have used sophisticated, complex tasks. For example, he and two of his students (also workshop participants) Kevin Dunbar and Jeff Shrager asked participants in a series of experiments to program a device called the Big Trak, and studied the processes they used to solve this problem. The Big Trak was a battery-powered vehicle that could be programmed, via a keypad, to move according to instructions. One of the keys was labeled RPT. Participants had to discover its function.

Following in Herb Simon's footsteps, Klahr, with Dunbar and Schunn, characterized subjects' performance as a search in two problem spaces, one occupied by possible experiments, the other by hypotheses [17]. They found that one group of subjects (Theorists) preferred to work in the hypothesis space, proposing about half as many experiments as the second group (Experimenters). Almost all of the former's experiments were guided by a hypothesis, whereas the latter's were often simply exploratory.

Based on this and other work, Klahr proposed a possible general framework, or shareable toothbrush, for classifying the different kinds of cognitive studies. This general framework is based on multiple problem spaces, and whether the study was a general one, using an abstract task like the Big Trak, or domain-specific, like Nersessian's studies of Maxwell [18].

Dunbar, currently at McGill University and moving to Dartmouth in the fall, added to this general framework the idea of classifying experiments based on whether they were in vitro (controlled laboratory experiments) or in vivo (case studies). Computational simulations can be based on either in vivo or in vitro studies. A system for classifying studies of scientific discovery might begin with a 2x2 matrix. Big Trak is an example of an in vitro technique; the work on Maxwell described by Nersessian,

on Faraday by Tweney, and on nuclear fission by Andersen, are examples of in vivo work. The three in vivo research programs did not explicitly distinguish between hypothesis and experiment spaces, but the practitioners studied generated both hypotheses and experiments.

The rest of this paper will feature highlights from the workshop that will force us to expand and transform this classification scheme (see Table I). Dunbar's work has iterated between in vivo and in vitro studies. The value of in vitro work is the way in which it allows for control and isolation of factors—like the way in which the possibility of error encourages experimental participants to adopt a confirmatory heuristic [19].

Dunbar thinks it is important to compare such findings with what scientists actually do. He has conducted a series of in vivo studies of molecular biology laboratories [20, 21]. Group studies have the heuristic value of forcing people to explain their reasoning. Regarding error, the molecular biologists had evolved special controls to check each step in a complex procedure in order to eliminate error. Dunbar ran an in vitro study in which he found that undergraduate molecular biology students would also employ this kind of control on a task that simulated the kind of reasoning used in molecular biology [22]. Dunbar's work shows the importance of iterating between in vitro and in vivo studies.

Schunn and his colleagues were interested in how scientists deal with unexpected results, or anomalies. In one study, he videotaped two astronomers interacting over a new set of data concerning the formation of ring galaxies. Schunn found that these researchers noticed anomalies as much as expected results, but paid more attention to the anomalies. The researchers developed hypotheses about the anomalies and elaborated on them visually, whereas they used theory to elaborate on expected results. When the two astronomers discussed the anomalies, they used terms like 'the funky thing' and 'the dipsy-doodle', staying at a perceptual rather than a theoretical level. Schunn's astronomers were working neither in the hypothesis nor experimental space; instead, they were working in a space of possible visualizations dependent on their domain-specific experience.

Hanne Andersen, from the University of Copenhagen, described the use of a family resemblance view of taxonomic concepts for understanding the dynamics of conceptual change. She noted that the family resemblance account has been criticized for not being able to distinguish sufficiently between different concepts, the problem of wide-open texture. This limitation could be resolved by including dissimilarity as well as similarity between concepts and by focusing on taxonomies instead of individual concepts. Anomalies can be viewed as violations of taxonomic principles that then lead to conceptual change. Andersen applied this approach to the discovery of nuclear fission, finding that early models of disintegration and atomic structure were revised in light of anomalous experimental results of this taxonomic kind

Shrager, affiliated with the Department of Plant Biology, Carnegie Institution of Washington, and the Institute for the Study of Learning and Expertise, did a reflective study of his own socialization into phytoplankton molecular biology. In the beginning, he had to be told about every step, even when there were explicit instructions; he needed an extensive apprenticeship. As his knowledge grew, he noted that it was "somewhere between his head and his hands." As his skill developed, he was able to take some of his attention off the immediate task at hand and understand the purpose of the procedures he was using. On at least one occasion, this came together in the "blink of an eye." The cognitive framework he found most useful was his own tooth-

brush: view application [23]. To his surprise, Shrager found that, "What passes for theory in molecular biology is the same thing that passes for a manual in car mechanics." He found less of a need to keep reflective notes in his diary as he became more proficient, though he continued to record the details of experiments, where particular materials were stored and all the other procedural details that are vital to a molecular biologist. He commented that, "if you lose your lab notebook, you're hosed."

Gooding indicated that more abstract computational models of the spherical horse variety have not worked well for him. For him, "the beauty is in the dirt." In collaboration with Tom Addis, a computer scientist, he evolved a detailed, computational scheme for representing Faraday's experiments, hypotheses and construals [24]. Gooding thought that communication ought to be added to the matrix proposed by Klahr and Dunbar (See Table 1).

Paul Thagard, from the University of Waterloo, has been gathering ideas from leaders in the field about what it takes to be a successful scientist. According to Herb Simon, one should not work on what everyone else is working on and one needs to have a secret weapon, in his case, computational modeling. As part of a case study, Thagard interviewed a microbiologist, Patrick Lee, who accidentally discovered that a common virus has potential as a treatment for cancer. The discovery was the result of a "stupid" experiment in viral replication done by one of Lee's graduate students. The "stupid" experiment produced an anomalous result that eventually led to the generation of a new hypothesis about the virus' ability to kill cancer cells. This chain of events is an example of abductive hypothesis formation, in which hypotheses are generated and evaluated in order to explain data. Once a hypothesis was generated that fit the data, researchers used deduction to arrive at the hypothesis that the virus could kill cancer cells. Thagard raises the questions of how one decides what experiments to do and how one determines what is a good experiment. These questions are a critical part of the cognitive processes involved in discovery. Thagard is also looking at the role of emotions in scientific inquiry, in judgments about potential experiments, in reactions to unexpected results, and in reactions to successful experiments (Thagard's model of emotions and science: http://cogsci/uwaterloo.ca). Thagard suggested adding a space of questions to the Klahr framework.

Robert Rosenwein, a sociologist at Lehigh, presented an in vitro simulation of science (SCISIM) that comes close to an in vivo environment [25]. Students in a class like Gorman's Scientific and Technological Thinking (http://128.143.168.25/classes /200R/tcc200rf00.html) take on a variety of social roles in science. Some work in competing labs, others run funding agencies, still others run a journal and a newsletter. The students in the labs try to get funding for their experiments, and then publish the results. They do not do the kinds of fine-grained experimental processes done by participants in Big Trak; instead, they choose the variables they want to combine in an experiment, select a level of precision, and are given a result. Experiments cost 'simbucks' and salaries have to be paid, so there is continual pressure to fund the lab. There is a group of independent scientists as well, who have to decide which line of research to pursue. SCISIM adds another column to the matrix, for simulation of pursuit decisions. Pursuit decisions concern which research program to seek funding for (See Table 1).

Such decisions are usually made within a network of enterprises. Marin Simina, a cognitive scientist at Tulane, described a computational simulation of Alexander Graham Bell's network of enterprises. Howard Gruber coined the term 'network of enter-

prises' to describe the way in which Darwin pursued multiple projects that eventually played a synergistic role in his theory of evolution [26]. Similarly, Alexander Graham Bell had two major enterprises in 1873: making speech visible to the deaf, and sending multiple messages down a single wire. These enterprises were synthesized in his patent for a speaking telegraph, which focused on the type of current that would have to be used to transmit and receive speech [27, 28].

Simina created a program called ALEC, which simulated the discovery Bell made on June 2, 1875. At that time, Bell's primary goal was to reach fame and fortune by solving the problem of multiple telegraphy, Bell had suspended the goal of transmitting speech because his mental model for a transmitter contained an indefinite number of metal reeds—it was not clear how it could be built. On June 2, 1877, a single tuned reed transmitted multiple tones with sufficient volume to serve as a transmitter for the human voice. Bell was not seeking this result; he wanted the reed to transmit only a single tone. But this serendipitous result allowed him to activate his suspended goal and instruct Watson to build the first telephone [29]. ALEC was able to simulate the process of suspending the goal and how Bell was primed to reactivate it by a result.

5 Collaboration and Invention

Gary Bradshaw, a cognitive scientist at Mississippi State and a collaborator with Herb Simon, talked about "stepping off Herb's shoulders into his shadow." In a study of the Wright Brothers, he adapted Klahr's framework to invention, creating three spaces: function, hypothesis and design [30]. One of the major reasons the Wrights succeeded where others failed was that the brothers decomposed the problem into separate functions—like vertical lift, horizontal stability, and turning. Other inventors worked primarily in a design space, adding features like additional wings without the careful functional analysis done by the Wrights. This suggests that function and design spaces ought to be added for inventors (see Table 1).

To see how well his framework of invention work-spaces held up, Bradshaw tried another case—the rocket boys from West Virginia, immortalized in a book by Homer Hickam [31], and in the film *October Sky* [32]. Their problem of rocket construction could be decomposed into multiple spaces, but a complete factorial of all the possible variations would come close to two million cells, so they could not follow the strategy called Vary One Thing at a Time (VOTAT) —they did not have the resources. Although the elements of the rocket construction were not completely separable, they tested some variables in isolation, such as fuel mixtures in bottles. They also did careful post-launch inspection, and used theory to reduce the problem space; for example, they used calculus to derive their nozzle shape. They built knowledge as they went along, taking good notes. Team members also took different roles—one was more of a scientist, another more of an engineer and project manager.

Tweney argued from his own experience that the rocket system was much less decomposable than suggested by Bradshaw's analysis and that the West Virginia group seemed to hit upon some serendipitous decompositions. Tweney's rocket group was stronger in chemistry, so they used theory to create the fuel, and copied the nozzle design. Both were post-Sputnik groups active during the late 1950's, although Tweney insists that his was a less serious "rocket boy" group than the one studied by Hickam.

Mehalik, a Systems Engineer at the University of Virginia, developed a framework which combined Hutchins' analysis of distributed cognition 'in the wild' [12], with three states or stages in actor networks.

1. A top-down state in which one actor or group of actors controls the research program and tells others what to do.
2. A trading zone state in which no group of actors has a comprehensive view, but all are connected by a boundary object that each sees differently. Peter Galison uses particle detectors as an example of this sort of boundary object [34],
3. A shared representation state in which all actors have a common perspective on what needs to be accomplished, even if there is still some division of labor based on skills, aptitude and expertise.

Mehalik applied this framework to the invention of an environmentally sustainable furniture fabric by a global group. This network began with a shared mental model based on an analogy to nature, then struggled to settle into a stable trading zone in which participants would trade economic benefits and prestige. The resulting fabric has won almost a dozen major environmental awards and is seen as a leading example of innovative environmental design.

Klahr suggested that Mehalik's research might add another dimension to his overall framework: capturing work in groups and teams. It might be possible to take each of the major actants studied by Mehalik, look at what spaces they worked in, then show links between them and their different activities. Tweney raised an important question about distributed cognition—could intra-individual cognition be modeled in a way similar to inter-individual cognition by including the three-state framework?

Michael Hertz, from the University of Virginia, developed a tool for determining causal attribution, and applied it to Monsanto's initially unsuccessful introduction of GMO's into Europe. The tool did not allow Hertz to identify a primary cause, but it did reduce the complexity of the decision space for students studying the Monsanto case and trying to determine who or what was at fault. Shrager suggested implementing this tool in an Echo network that would incorporate interaction with the decision-makers themselves. Bernie Carlson raised the question of when it is useful to quantify certain decision situations, again relating to the theme of the balance between using a tool to help reduce complexity in a decision situation while still maintaining contextual validity. Ryan Tweney raised the issue of using Hertz's framework in a predictive sense—the dynamic complexity of the situation may be too difficult to make predictions; however, prediction is what a company such as Monsanto may be most interested in. Hertz responded by saying that the act of trying to identify causes has heuristic value, especially if a tool helps Monsanto distinguish between the relative role of factors it can influence and factors that are largely beyond its control. Decision aids and simulations simplify complex situations; decision-makers need to remember that these simplifications may not accurately reflect all important aspects of the underlying situation, including complex, dynamic interactions among variables.

Thomas Hughes, a historian of technology, talked about his analysis of collective invention in large-scale systems like the development of the Atlas and Polaris missiles [35]. He extolled the virtues of systems management techniques and the benefits of isolating scientists from bureaucracy. Project management and oversight functions change with the size of the group and management becomes more explicitly needed with larger groups. Without sufficient oversight, large projects can be too diffuse and

inefficient. Dunbar suggested that having this kind of systems management was one reason why the privately funded Celera outperformed the publicly funded Human Genome Project.

William (Chip) Levy, from the Department of Neurosurgery at the University of Virginia, described a neural network that models results of an implicit learning experiment. He uses the model as an illustration of how variability can be an adaptive property in biological terms. Complex systems, like brains and like neural network models, benefit from the random fluctuations of noise. Eliminating variability in these systems would sacrifice too much memory capacity. Variability exists both within and between individuals.

Levy's research highlights the role of tacit knowledge in discovery and invention. Sociologists of science and technology emphasize the tacit dimension [36, 37]. There is a growing cognitive literature on implicit knowledge in psychology [38, 39], but this literature does not connect directly to discovery and invention. Several conference participants mentioned tacit knowledge. Robert Matthews, a cognitive psychologist at Louisiana State and one of the leading researchers on implicit learning [40], predicted that Dunbar's scientists would be unable to explain why they did what they did. Dunbar responded that the scientists' after-the-fact stories about how they did what they did had nothing to do with their actual processes. Schunn noted Karmiloff-Smith's three stages of learning, in which the second stage means you can do something without being able to explain it, and the third stage involves reflection [41]. The way to become aware of one's implicit knowledge is to watch oneself, which can interfere with performance.

Maria Ippolito, from the University of Alaska, compared the creative process exhibited in the writings of Virginia Woolf to that used by scientists. Ippolito offered Woolf as an example of a scientific thinker in a more general sense and constructed a multi-dimensional database using Woolf's writings. Through the examination of Woolf's development as a writer, Ippolito investigated the psychological processes of creative problem solving, including heuristics, scripts and schemata, development of expertise, and search of unstructured problem spaces.

Elke Kurz, from the University of Tübingen, commented on two studies in which she observed the softening of often-perceived boundaries between cognitive-historical case study analysis and in-laboratory analyses. She examined how scientists and mathematicians used different representational systems, such as variant forms of Calculus, when problem solving. These differences can be traced to historical developments in the different scientific fields. Such historical developments invite historical case analysis as a necessary part of the study of the conceptual resources these different scientists possessed. Kurz also replicated experiments involving perception of size constancy that had been done earlier by Brunswik. During the attempts at replication, Kurz noted how Brunswik needed to constrain the participants' agency into forms that Brunswik found tolerable in the context of his experiment. Kurz stated the construction of this context of acceptable agency is a process worth studying using historical case methods, again complementing the in-laboratory style of investigation. Finally, Kurz reported on the difficulties of attempting a replication of a previous experiment because of the changes in many contextual events between the original experiment and the replicated experiment. This situation again invites the crossing of any perceived boundary between the case study and in-laboratory approaches.

6 Lessons Learned

The workshop illustrates that toothbrushes can be shared. The example we used in this paper was the Simon/Klahr multiple spaces framework. Table 1 summarizes the potential spaces identified in the workshop.

Table 1. Different search spaces identified by participants in the workshop. Asterisks denote computational simulations, a kind of 'spherical horse' that can be based on either in vivo or in vitro studies. Italics denote spaces that are unique to invention.

Search Spaces	In Vitro	In Vivo
Hypotheses	Big Trak, SciSim	Maxwell, Faraday
Experiments	Big Trak, SciSim	Maxwell, Faraday
Pursuit	SciSim	ALEC*
Communication	SciSim	Faraday
Embodied knowledge		Faraday, Shrager
Taxonomies		Nuclear fission
Visualizations		Galatea*, Schunn's astronomers
Questions		Patrick Lee
Links in a social network		Hughes, Mehalik
Function		Wright brothers, rocket boys
Design		Wright brothers, rocket boys

The problem with this framework is that each study seemed to suggest the need for yet another space. There is not always a clear line of demarcation between spaces. For example, SciSim incorporates in vivo cases, which means that it can exist in a kind of gray zone between in vitro and in vivo. Visualizations can be thought experiments, ways of seeing the data, and mental models of a device or even of a social network. Despite its shortcomings, this framework has heuristic value, both for organizing research already done and for suggesting directions for future work. For example, only Bradshaw has worked with function and design spaces, and there is no in vitro work on invention

Mehalik's work demonstrated the need for mapping movements among spaces across individuals over time. What would happen if we added time-scale to the framework? Schun suggested that visualizations happen most rapidly, with experiments and hypotheses taking longer, and taxonomies even longer.[3] Hughes and Mehalik remind us that time-scale is partly dependent on the extent to which each of these activities depends on network-building.

This framework is also general enough to facilitate comparisions between discovery, invention and artistic creation, as Ippolito noted. More comparisions of this sort are needed.

[3] Personal communication.

7 Future of Cognitive Studies of Science and Technology

Bruce Seely, a historian of technology on rotation at the NSF's Science and Technology Studies program, felt that the workshop showed how cognitive studies of science and technology had grown in sophistication. highlighting the creators of new knowledge in ways that complemented studies of users by other STS disciplines.

Tiha von Ghyczy, representing the Strategic Institute of the Boston Consulting Group, noted that managers are happy to use any toothbrush that will help them improve their business strategies, and they are also more concerned about practical results than methodological foundations. Still, he felt that managers would find lessons from the workshop interesting. Strategies have a very short half-life; a successful strategy is quickly imitated by competitors. Therefore, original thinking is essential for business survival.

Besides business strategy and science-technology studies, a cognitive approach to invention and discovery should also inform work in 'mainstream' cognitive science. Theories and frameworks need to be able to deal in a rigorous way with the shared and distributed character of scientific and technological problem solving, and also its tacit dimension. We hope this workshop will suggest the outlines more sophisticated theories and models might take. Ideally, anyone doing a computational model or decision-aid for discovery would base it on one or more fine-grained case studies. Tweney and Dunbar have had particularly good success combining in vitro and in vivo approaches. We hope this workshop will encourage more collaborations between those trained in spherical-horse approaches and those capable of going deeply into the details of particular discoveries and inventions.

References

1. Christensen, C.M., *The innovator's dilemma: When new technologies cause great firms to fail*. 1997, Boston: Harvard Business School Press.
2. Evans, P. and T.S. Wurster, *Blown to bits: How the new economics of information transforms strategy*. 2000, Boston: Harvard Business School Press.
3. Nonaka, I. and H. Takeuchi, *The knowledge-creating company: how Japanese companies create the dynamics of innovation*. 1995, New York: Oxford University Press.
4. Gorman, M.E., *et al.*, *Alexander Graham Bell, Elisha Gray and the Speaking Telegraph: A Cognitive Comparison.* History of Technology, 1993. **15**: p. 1-56.
5. Shrager, J. and P. Langley, *Computational Models of Scientific Discovery and Theory Formation*. 1990, San Mateo, CA: Morgan Kaufmann Publishers, Inc.
6. Simon, H.A., Langley, P. W., & Bradshaw, G., *Scientific discovery as problem solving.* Synthese, 1981. **47**: p. 1-27.
7. Slezak, P., *Scientific discovery by computer as empirical refutation of the Strong Programme.* Social Studies of Science, 1989. **19(4)**: p. 563-600.
8. Langley, P., Simon, H. A., Bradshaw, G. L., & Zykow, J. M. *Scientific Discovery: Computational Explorations of the Creative Processes*. 1987, Cambridge: MIT Press.

9. Bradshaw, G.L., Langley, P., & Simon, H. A., *Studying scientific discovery by computer simulation.* Science, 1983. **222**: p. 971-975.
10. Vera, A.H. and H.A. Simon, *Situated action: A symbolic interpretation.* Cognitive Science, 1993. **17**(1): p. 7-48.
11. Greeno, J.G. and J.L. Moore, *Situativity and symbols: Response to Vera and Simon.* Cognitive Science, 1993. **17**: p. 49-59.
12. Hutchins, E., *Cognition in the Wild*. 1995, Cambridge, MA: MIT Press.
13. Norman, D.A., *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. 1993, New York: Addison Wesley.
14. Donald, M., *Origins of the modern mind: Three stages in the evolution of culture and cognition*. 1991, Cambridge, UK: Harvard.
15. Nersessian, N., *How do scientists think? Capturing the dynamics of conceptual change in science*, in *Cognitive Models of Science*, R.N. Giere, Editor. 1992, University of Minnesota Press: Minneapolis. p. 3-44.
16. Suchman, L.A., *Plans and Situated Actions: The Problem of Human-Machine Interaction*. 1987, Cambridge: Cambridge University Press.
17. Klahr, D., *Exploring Science: The cognition and development of discovery processes*. 2000, Cambridge, MA: MIT Press.
18. Klahr, D. and H.A. Simon, *Studies of Scientific Discovery: Complementary approaches and convergent findings.* Psychological Bulletin, 1999. **125**(5): p. 524-543.
19. Gorman, M.E., *Simulating Science: Heuristics, Mental Models and Technoscientific Thinking*. Science, Technology and Society, ed. T. Gieryn. 1992, Bloomington: Indiana University Press. 265.
20. Dunbar, K., *How scientists really reason: Scientific reasoning in real-world laboratories*, in *The nature of insight*, R.J. Sternberg and J. Davidson, Editors. 1995, MIT Press: Cambridge, MA. p. 365-396.
21. Dunbar, K., *How scientists think*, in *Creative Thought*, T.B. Ward, S.M. Smith, and J. Vaid, Editors. 1997, American Psychological Association: Washington, D.C.
22. Dunbar, K. *Scientific reasoning strategies in a simulated molecular genetics environment*. in *Program of the Eleventh Annual Conference of the Cognitive Science Society*. 1989. Ann Arbor, MI: Lawrence Erlbaum Associates.
23. Shrager, J., *Commonsense perception and the psychology of theory formation*, in *Computational Models of Scientific Discovery and Theory Formation*, J. Shrager, & Langley, P., Editor. 1990, Morgan Kaufmann Publishers, Inc.: San Mateo, CA. p. 437-470.
24. Gooding, D.C. and T.R. Addis, *Modelling Faraday's experiments with visual functional programming 1: Models, methods and examples*, . 1993, Joint Research Councils' Initiative on Cognitive Science & Human Computer Interaction Special Project Grant #9107137.
25. Gorman, M. and R. Rosenwein, *Simulating social epistemology.* Social Epistemology, 1995. **9**(1): p. 71-79.
26. Gruber, H., *Darwin on Man: A Psychological Study of Scientific Creativity*. 2nd ed. 1981, Chicago: University of Chicago Press.
27. Gorman, M.E. and J.K. Robinson, *Using History to Teach Invention and Design: The Case of the Telephone.* Science and Education, 1998. **7**: p. 173-201.

28. Simina, M., *Enterprise-directed reasoning: Opportunism and deliberation in creative reasoning*, in *Cognitive Science*. 1999, Georgia Institute of Technology: Atlanta, GA.
29. Gorman, M.E., *Transforming nature: Ethics, invention and design*. 1998, Boston: Kluwer Academic Publishers.
30. Bradshaw, G., *The Airplane and the Logic of Invention*, in *Cognitive Models of Science*, R.N. Giere, Editor. 1992, University of Minnesota Press: Minneapolis. p. 239-250.
31. Hickam, H.H., *Rocket boys: A memoir*. 1998, New York: Delacorte Press. 368.
32. Gordon (producer), C. and J. Johnston (director), *October Sky*, . 1999, Universal Studios: Universal City, CA.
33. Tweney, R.D., *Scientific Thinking: A cognitive-historical approach*, in *Designing for Science: Implications for everyday, classroom, and professional settings*, K. Crowley, C.D. Schunn, and T. Okada, Editors. 2001, Lawrence Earlbaum & Associates: Mawah, NJ. p. 141-173.
34. Galison, P.L., *Image and logic: A material culture of microphysics*. 1997, Chicago: University of Chicago Press.
35. Hughes, T.P., *Rescuing Prometheus*. 1998, New York: Pantheon books.
36. Collins, H.M., *Tacit knowledge and scientific networks*, in *Science in context: Readings in the sociology of science*, B. Barnes and D. Edge, Editors. 1982, The MIT Press: Cambridge, MA.
37. Mackenzie, D. and G. Spinardi, *Tacit knowledge, weapons design, and the uninvention of nuclear weapons.* American Journal of Sociology, 1995. **101**(1): p. 44-99.
38. Berry, D.C., ed. *How implicit is implicit learning?* . 1997, Oxford University Press: Oxford.
39. Dienes, Z. and J. Perner, *A Theory of Implicit and Explicit Knowledge.* Behavioral and Brain Sciences, 1999. **22**(5).
40. Matthews, R.C. and L.G. Roussel, *Abstractness of implicit knowledge: A cognitive evolutionary perspective*, in *How implicit is implicit learning?*, D.C. Berry, Editor. 1997, Oxford University Press: Oxford. p. 13-47.
41. Karmiloff-Smith, A., *From meta-process to conscious access: Evidence from children's metalinguistic and repair data.* Cognition, 1986. **23**(2): p. 95-147.

Clipping and Analyzing News Using Machine Learning Techniques

Hans Gründel, Tino Naphtali, Christian Wiech,
Jan-Marian Gluba, Maiken Rohdenburg, and Tobias Scheffer

SemanticEdge, Kaiserin-Augusta-Allee 10-11, 10553 Berlin, Germany
{hansg, tinon, christianw, jang, scheffer}@semanticedge.com

Abstract. Generating press clippings for companies manually requires a considerable amount of resources. We describe a system that monitors online newspapers and discussion boards automatically. The system extracts, classifies and analyzes messages and generates press clippings automatically, taking the specific needs of client companies into account. Key components of the system are a spider, an information extraction engine, a text classifier based on the Support Vector Machine that categorizes messages by subject, and a second classifier that analyzes which emotional state the author of a newsgroup posting was likely to be in. By analyzing large amount of messages, the system can summarize the main issues that are being reported on for given business sectors, and can summarize the emotional attitude of customers and shareholders towards companies.

1 Introduction

Monitoring news paper or journal articles, or postings to discussion boards is an extremely laborious task when carried out manually. Press clipping agencies employ thousands of personnel in order to satisfy their clients' demand for timely and reliable delivery of publications that relate to their own company, to their competitors, or to the relevant markets. The internet presence of most publications offers the possibility of automating this filtering and analyzing process. One challenge that arises is to analyze the content of a news story well enough to judge its relevance for a given client. A second difficulty is to provide appropriate overview and analyzing functionality that allows a user to keep track of the key content of a potentially huge amount of relevant publications.

Software systems that spider the web in search of relevant information, and extract and process found information are usually referred to as *information agents* [16,2]. They are being used, for instance, to find interesting web sites or links [19,12], or to filter news group postings (*e.g.,* [26]). One attribute of information agents is how they determine the relevance of a document to a user.

Content-based recommendation systems (*e.g.,* [1]) judge the interestingness of a document to the user based on the content of other documents that the user has found interesting. By contrast, collaborative filtering approaches (*e.g.,* [13]),

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 87–99, 2001.

draw conclusions based on which documents other users with similar preferences have found interesting. In many applications, it is not reasonable to ask the user to elaborate his or her preferences explicitly. Therefore, information agents often try to *learn* a function that expresses user interest from user feedback; *e.g.,* [26,18]. By contrast, a user who approaches a press clipping agency usually has specific, elaborated information needs.

The problem of identifying predetermined relevant information in text or hypertext documents from some specific domain is usually referred to as *information extraction* (IE) (*e.g.,* [4,3]). In the news clipping context, several instances of the information extraction problem occur. Firstly, press articles have to be extracted from HTML pages where they are usually embedded between link collections, adverts, and other surrounding text. Secondly, named entities such as companies or products have to be identified and extracted and, thirdly, meta-information such as publication dates or publishers need to be found.

While first IE algorithms were hand-crafted sets of rules (*e.g.,* [7]), algorithms that learn extraction rules from hand-labeled documents (*e.g.,* [8,14,6]) have now become standard. Unfortunately, rule-based approaches sometimes fail to provide the necessary robustness against the inherent variability of document structure, which has led to the recent interest in the use of Hidden Markov Models (HMMs) [25,17,21,23] for this purpose.

In order to identify whether the content of a document matches one of the categories the user is interested in and to summarize the subjects of large amounts of relevant documents, classifiers that are learned from hand-labeled documents (*e.g.,* [24,11]) provide a means of categorizing a document's content that reaches far beyond key word search. Furthermore, it can be interesting to determine the emotional state [9] of authors of postings about a company or product.

In this paper, we discuss a press clipping information agent that downloads news stories from selected news sources, classifies the messages by subject and business sector, and recognizes company names. It then generates customized clippings that match the requirement of clients. We describe the general architecture in Section 2, and discuss the machine learning algorithms involved in Section 3. Section 4 concludes.

2 Publication Monitoring System

Figure 1 sketches the general architecture of the system. A user can configure the information service by providing a set of preferences. These include the names of all companies that he or she would like to monitor, as well as all business areas (*e.g.,* biotechnology, computer hardware) of interest The spider cyclically downloads a set of newspapers, journals, and discussion boards. The set of news sources is fixed in advance and not depending on the users' choices. All downloaded messages are recorded in a news database after the extraction engine has stripped the HTML code in which the message is embedded (header and footer parts as well as HTML tags, pictures, and advertisements).

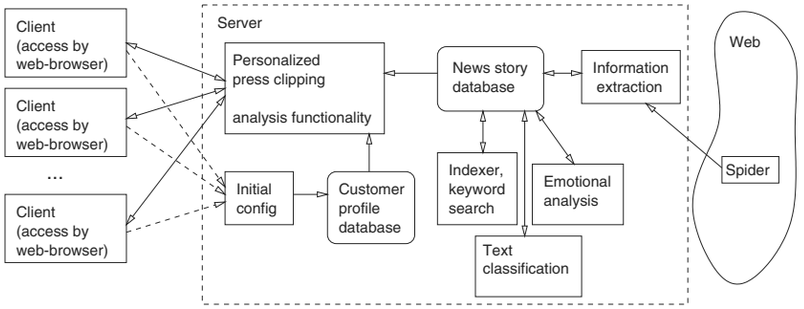


Fig. 1. Overview of the SemanticEdge Publication Monitoring System

The spider developed by SemanticEdge is configured by providing a set of patterns which all URLs that are to be downloaded have to match. Typically, online issues of newspapers have a fairly fixed site structure and only vary the dates and story numbers in the URLs daily. Depending on the difficulty of the site structure, configuring the spider such that all current news stories but no advertisements, archives, or documents that do not directly belong to the newspaper are downloaded, requires between one and four hours.

Text classifier, named entity recognizer, and emotional analyzer operate on this database. The text classifier categorizes all news stories and newsgroup postings whereas the emotional analyzer is only used for newsgroup postings; it classifies the emotional state that a message was likely to be written in. For each client company, a customized press clipping is generated, including summarization and visualization functionality.

The press clipping consists of a set of dynamically generated web pages that a user can view in a browser after providing a password. The system visualizes the number of publications by source, by subject, and by referred company. For each entry, an emotional score between zero (very negative) and one (very positive) is visualized as a red or green bar, indicating the attitude of the article (Fig. 2), or the set of summarized articles. Figure 2 shows the list of all articles relevant to a client, Figure 3 shows the summary mode in which the system summarizes all articles either from one news source, or about one company, or related to one business sector per line. The average positive or negative attitude of the articles summarized in one line is visualized by a red or green bar.

Several diagrams visualize the frequency of referrals to business sectors or individual companies and the average expressed attitude (Figure 4).

3 Intelligent Document Analysis

Document analysis consists of information extraction (including recognition of named entities), subject classification, and emotional state analysis.

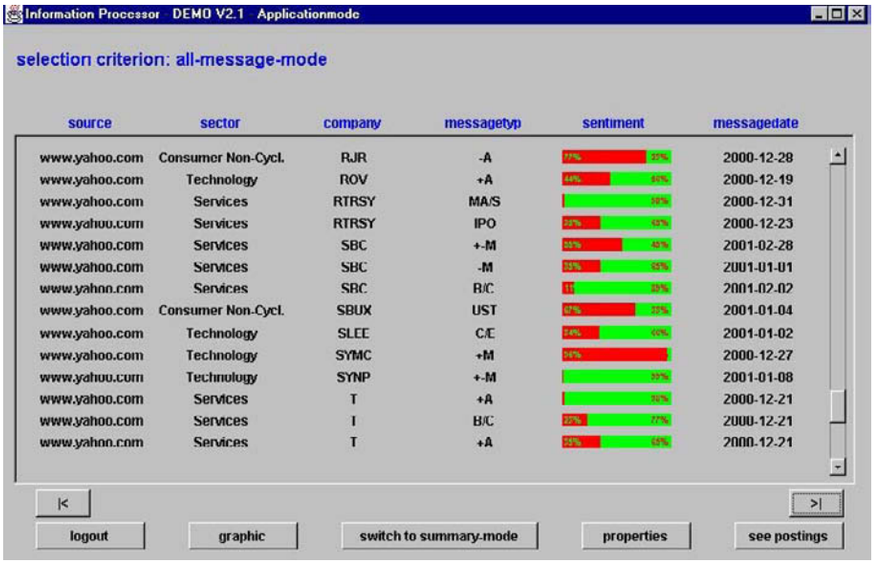


Fig. 2. Press clipping for client company: message overview

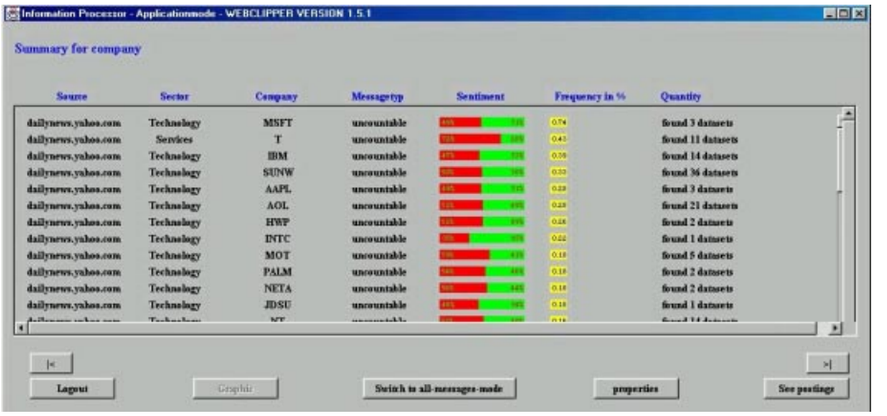


Fig. 3. Press clipping for client company: company summary

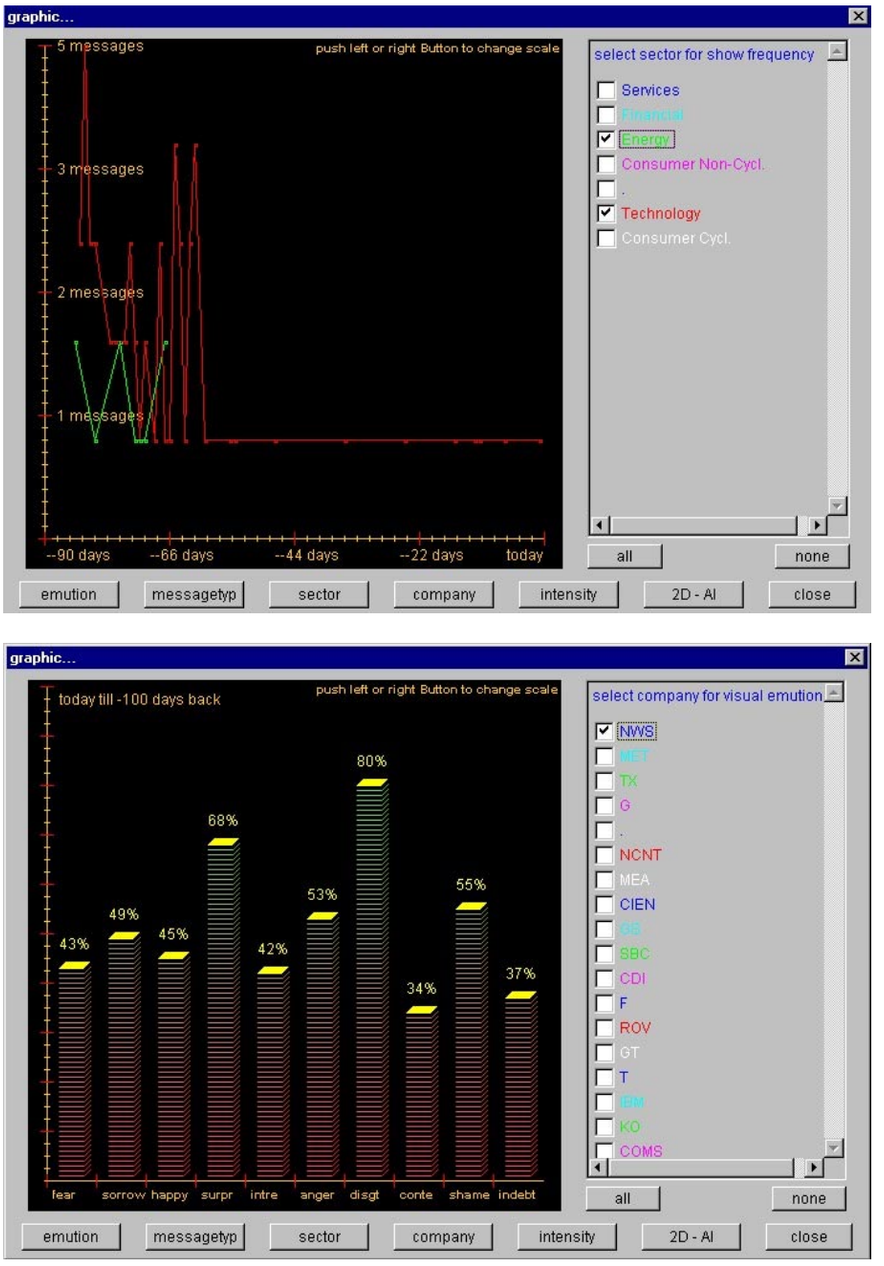


Fig. 4. Top: frequency of messages related to business sectors. Bottom: expressed emotional attitude toward companies

3.1 Information Extraction

Two main paradigms of information extraction agents which can be trained from hand-labeled documents exist; algorithms that learn extraction rules (*e.g.,* [8,14, 6]) and statistical approaches such as Markov models [25,17], partially hidden Markov models [21,23] and conditional random fields [15].

Rule base information extraction algorithms appear to be particularly suited to extract text from pages with a very strict structure and little variability between documents. In order to learn how to extract the text body from the HTML page of a Yahoo! message board, the proprietary rule learner that we use needs only one example in order to identify where, in the document structure, the information to be extracted is located. We can then extract text bodies from other messages with equal HTML structure with an accuracy of 100%.

Many other information extraction tasks, such as recognizing company names, or stock recommendations, rule based learners do not provide enough robustness to deal with the high variability of natural language. Hidden Markov models (HMMs) (see, [20] for an introduction) are a very robust statistical method for analysis of temporal data. An HMM consists of finitely many states $\{S_1, \ldots, S_N\}$ with probabilities $\pi_i = P(q_1 = S_i)$, the probability of starting in state S_i, and $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$, the probability of a transition from state S_i to S_j. Each state is characterized by a probability distribution $b_i(O_t) = P(O_t | q_t = S_i)$ over observations. In the information extraction context, an observation is typically a token. The information items to be extracted correspond to the n target states of the HMM. Background tokens without label are emitted in all HMM states which are not one of the target states.

HMM parameters can be learned from data using the Baum-Welch algorithm. When the HMM parameters are given, then the model can be used to extract information from a new document. Firstly, the document has to be transformed into a sequence of tokens; for each token, several attributes are determined, including the word stem, part of speech, the HTML context, attributes that indicates whether the word contains letter, digits, starts with a capital letter and other attributes. Thus, the document is transformed into a sequence of attribute vectors.

Secondly, the forward-backward algorithm [20] is used to determine, for each token, the most likely state of the HMM that it was emitted in. If, for a given token, the most likely state is one of the background states, then this token can be ignored. If the most likely state is one of the target states and thus corresponds to one of the items to be extracted, then the token is extracted and copied into the corresponding database field.

In order to adapt the HMM parameter, a user first has to label information to be extracted manually in the example document. Such partially labeled documents form the input to the learning algorithm which then generates the HMM parameters. We use a variant of the Baum-Welch algorithm [23] to find the model parameters which are most likely to produce the given documents and are consistent with the labels added by the user.

Figure 5 shows the GUI of the SemanticEdge information extraction environment. HMM based and rule based learners are plugged into the system.

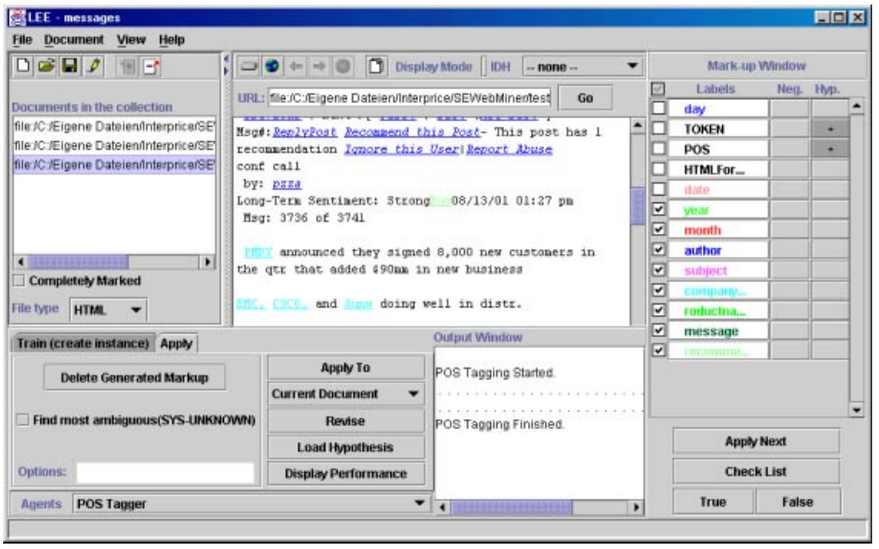


Fig. 5. GUI of the information extraction engine

For specialized information extraction tasks such as finding company names in news stories, specially tailored information extraction agents outperform more general approaches such as HMMs. For instance, most companies that are being reported about are listed at some stock exchange. To recognize these companies, we only need to maintain a dynamically growing database.

3.2 Subject Classification

For the subject classification step, we have defined a set of message subject categories (*e.g.,* IPO announcement, ad hoc message) and a set of business sector and markets categories. The resulting classifiers assign each message a set of relevant subjects and sectors.

The classifier proceeds in several steps. First, a text is tokenized and the resulting tokens are mapped to their word stems. We then count, for each word stem and each example text, how often that word occurs in the text. We thus transform each text into a feature vector, treating a text as a bag of words. Finally, we weight each feature by the inverse frequency of the corresponding word which has generally been observed to increase the accuracy of the resulting

classifiers (*e.g.,* [10,22]). This procedure maps each text to a point in a high-dimensional space.

The Support Vector Machine (SVM) [11] is then used to efficiently find a hyper-plane which separates positive from negative examples, such that the margin between any example and the plane is maximized. For each category we thus obtain a classifier which can then take a new text and map it to a negative or positive values, measuring the document's relevance for the category.

During the application phase, the support vector machine returns, for each category, a value of its decision function, that can range from large negative to large positive values. It is necessary to define a threshold value from which on a document is considered to belong to the corresponding category. There are several criteria by which this threshold can be set; perhaps the most popular is the precision recall breakeven point. The precision quantifies the probability of a document really belonging to a class given that it is predicted to lie in that class. Recall, on the other hand quantifies how likely it is that a document really belonging to a category is in fact predicted to be a member of that class by the classifier. By lowering the threshold value of the decision function we can increase recall and decrease precision, and vice versa. The point at which precision equals is often used as a normalized measure of the performance of classification and IR methods. Varying the threshold leads to precision and recall curves. Figure 6 shows the GUI of our text SVM-based categorization tool.

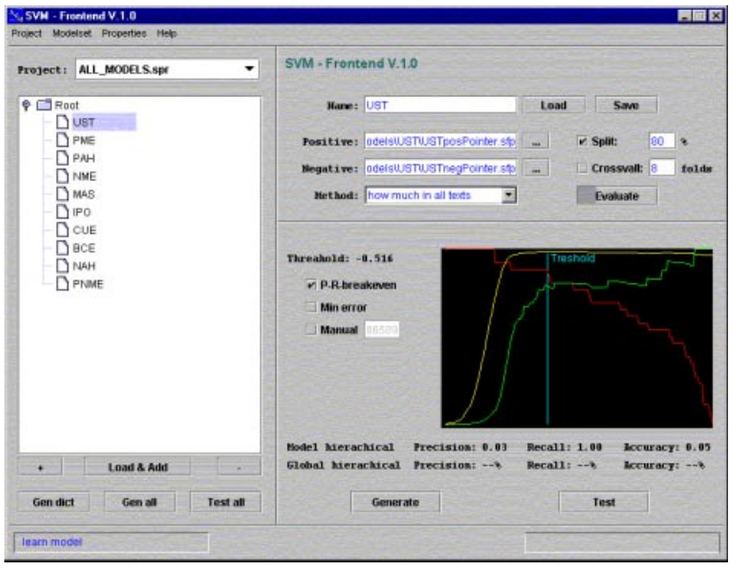


Fig. 6. GUI of the text classification engine

It is also possible to define the accuracy (the probability of the classifier making a correct prediction for a new document) as a performance measure. Unfortunately, many categories (such as IPO announcement) are so infrequent, that a classifier which in fact *never* predicts that a document does belong to this class can achieve an accuracy of as much as 99.9%. This renders the use of accuracy as a performance metric less suited than precision/recall curves.

For each category, we manually selected about 3000 examples, between 60 and 700 of these examples were positives. Figure 7 shows precision, recall, and accuracy of some randomly selected classes over the threshold value. The curves are based on hold-out testing on 20% of the data. Note that, for many of these classes such as xxx, the prior ratio of positive examples is extremely small (such as 1.4%). Specialized categories, such as "initial public offering announcement" can be recognized almost without error; "fuzzy" concepts like "positive marked news" impose greater uncertainties.

3.3 Emotional Classification

In psychology, a space of universal, culturally independent base emotional states have been identified according to the differential emotional theory (*e.g.,* [5,9]); ten clusters within this emotional space are generally considered base emotions. These are interest, happiness, surprise, sorrow, anger, disgust, contempt, shame, fear, guilt (Figure 4).

While it is typically impossible to analyze the emotional state of the author of a sober newspaper article, authors of newsgroup often do not conceal their emotions. Given a posting, we use an SVM to determine, for each of the ten emotional states, a score that rates the likelihood of that emotion for the author. We average the scores over all postings related to a company, or to a product, and visualize the result as in Figure 4. We can project emotional scores onto a "positive-negative" ray and visualize the resulting score as a red or green bar as in Figure 2.

We manually classified postings to discussion boards into positive, negative, and neutral for each of the ten base emotional states. Emotional classification of messages turned out to be a fairly noisy process; the judgment on the emotional content of postings usually varies between individuals. Unfortunately, we found no positive examples for disgust but between 2 and 21 positive and between 16 and 92 negative examples for the other states.

Figure 8 shows precision and recall curves for those emotions for which we found most positive examples, based on 10-fold cross validation. As we expected, recognizing emotions seems to be a very difficult task; in particular, from the small samples available. Still the recognizer performs significantly better than random guessing. Emotional classifiers with a rather high threshold can often achieve reasonable precision values. Also, in many cases in which human and classifier disagree, it is not easy to tell whether human or classifier are wrong.

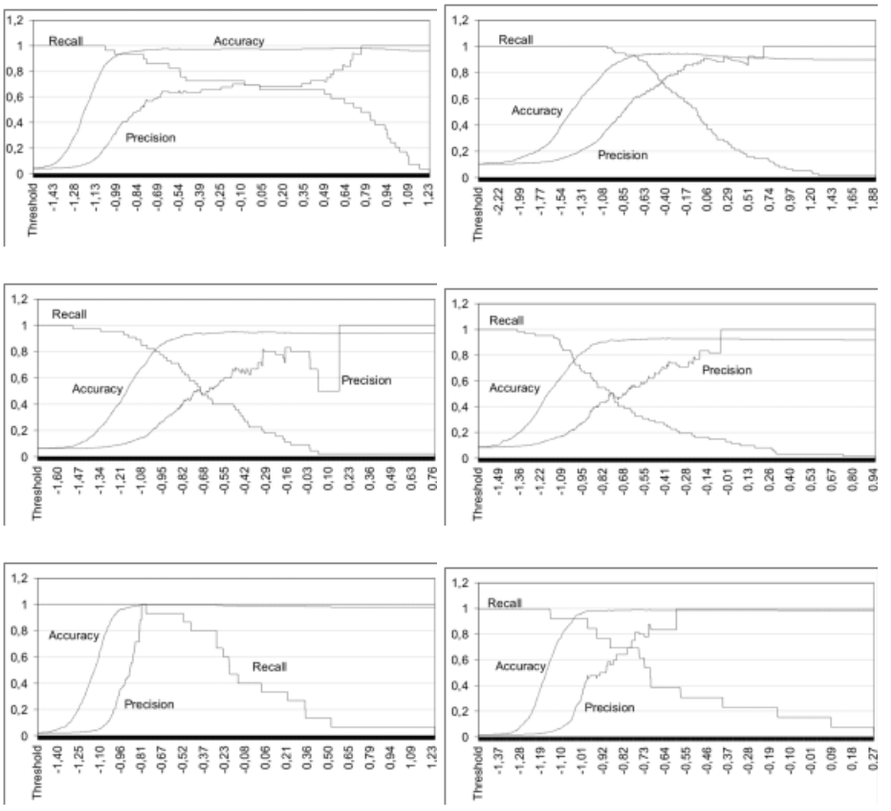


Fig. 7. Precision, recall, and accuracy for subject classification. First row: "US treasury", "mergers and acquisition"; second row: "positive" / "negative market and economy news"; third row: "initial public offering announcement", "currency and exchange rates".

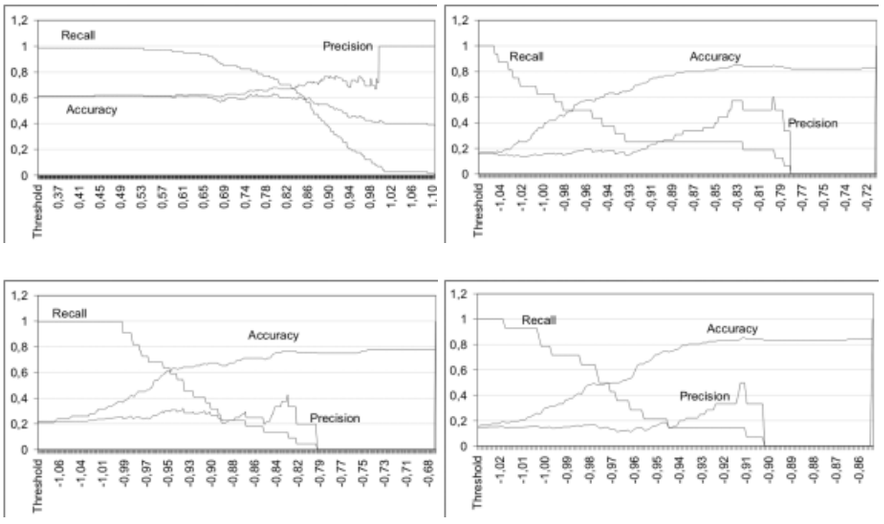


Fig. 8. Precision, recall, and accuracy for emotional classification. First row: positive versus negative, anger; second row: contempt, fear.

4 Conclusion

We describe a system that monitors online news sources and discussion boards, downloads the content regularly, extracts the document bodies, analyzes messages by content and emotional state, and generates customer-specific press clippings. A user of the system can specify his or her information needs by entering a list of company names (*e.g.,* the name of the own company and relevant competitors) and selecting from a set of message types and business sectors. Information extraction tasks are addressed by rule induction and hidden Markov modes; the Support Vector Machine is used to learn classifiers from hand-labeled data. The customer-specific news stories are listed individually, as well as summarized by several criteria. Diagrams visualize how frequently business sectors or companies are cited over time.

The resulting press clippings are generated in near real-time and fully automatically. This tool enables companies to keep track of how they are being perceived in news groups and in the press. It is also inexpensive compared to press clipping agencies. On the down side, the system is certain to miss all news stories that only appear in printed issues. Also, the classifier has a certain inaccuracy which imposes the risk of missing relevant articles. Of course, this risk is also present with press clipping agencies. Nearly all studied subject categories can be recognized very reliably using support vector classifiers.

References

1. A. Amrodt and E. Plaza. Cased-based reasoning: foundations, issues, methodological variations, and system approaches. *AICOM*, 7(1):39–59, 1994.
2. N. Belkin and W. Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.
3. Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew K. McCallum, Tom M. Mitchell, Kamal Nigam, and Seán Slattery. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118(1-2):69–113, 2000.
4. L. Eikvil. Information extraction from the world wide web: a survey. Technical Report 945, Norwegian Computing Center, 1999.
5. P. Ekman, W. Friesen, and P. Ellsworth. *Emotion in the human face: Guidelines for research and integration of findings*. Pergamon Press, 1972.
6. G. Grieser, K. Jantke, S. Lange, and B. Thomas. A unifying approach to html wrapper representation and learning. In *Proceedings of the Third International Conference on Discovery Science*, 2000.
7. Ralph Grishman and Beth Sundheim. Message understanding conference - 6: A brief history. In *Proceedings of the International Conference on Computational Linguistics*, 1996.
8. N. Hsu and M. Dung. Generating finite-state transducers for semistructured data extraction from the web. *Journal of Information Systems, Special Issue on Semistructured Data*, 23(8), 1998.
9. C. Izard. *The face of emotion*. Appleton-Century-Crofts, 1971.
10. T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, 1997.
11. T. Joachims. Text categorization with support vector machines. In *Proceedings of the European Conference on Machine Learning*, 1998.
12. Thorsten Joachims, Dayne Freitag, and Tom Mitchell. WebWatcher: A tour guide for the World Wide Web. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 770–777, San Francisco, August23–29 1997. Morgan Kaufmann Publishers.
13. J. Konstantin, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.
14. N. Kushmerick. Wrapper induction: efficiency and expressiveness. *Artificial Intelligence*, 118:15–68, 2000.
15. John Lafferty, Fernando Pereira, and Andrew McCallum. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.
16. P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7), 1994.
17. Andrew McCallum, Dayne Freitag, and Fernando Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.
18. A. Moukas. Amalthaea: Information discovery and filtering using a multiagent evolving ecosystem. In *Proceedings of the Conference on Practical Application of Intelligent Agents and Multi-Agent Technology*, 1996.
19. M Pazzani, J. Muramatsu, and D. Billsus. Syskill and webert: Identifying interesting web sites. In *Proceedings of the National Conference on Artificial Intelligence*, pages 54–61, 1996.

20. L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.
21. T. Scheffer, C. Decomain, and S. Wrobel. Active hidden markov models for information extraction. In *Proceedings of the International Symposium on Intelligent Data Analysis*, 2001.
22. T. Scheffer and T. Joachims. Expected error analysis for model selection. In *Proceedings of the Sixteenth International Conference on Machine Learning*, 1999.
23. Tobias Scheffer and Stefan Wrobel. Active learning of partially hidden markov models. In *Proceedings of the ECML/PKDD Workshop on Instance Selection*, 2001.
24. M. Sehami, M. Craven, T. Joachims, and A. McCallum, editors. *Learning for Text Categorization, Proceedings of the ICML/AAAI Workshop*. AAAI Press, 1998.
25. Kristie Seymore, Andrew McCallum, and Roni Rosenfeld. Learning hidden markov model structure for information extraction. In *AAAI'99 Workshop on Machine Learning for Information Extraction*, 1999.
26. B. Sheth. Newt: A learning approach to personalized information filtering. Master's thesis, Department of Electiric Engineering and Computer Science, MIT, 1994.

Towards Discovery of Deep and Wide First-Order Structures: A Case Study in the Domain of Mutagenicity

Tamás Horváth[1] and Stefan Wrobel[2]

[1] Institute for Autonomous intelligent Systems, Fraunhofer Gesellschaft, Schloß Birlinghoven, D-53754 Sankt Augustin,
tamas.horvath@fhg.de
[2] Otto-von-Guericke-Universität Magdeburg, IWS, P.O.Box 4120, D-39106 Magdeburg,
wrobel@iws.cs.uni-magdeburg.de

Abstract. In recent years, it has been shown that methods from Inductive Logic Programming (ILP) are powerful enough to discover new first-order knowledge from data, while employing a clausal representation language that is relatively easy for humans to understand. Despite these successes, it is generally acknowledged that there are issues that present fundamental challenges for the current generation of systems. Among these, two problems are particularly prominent: learning *deep clauses*, i.e., clauses where a long chain of literals is needed to reach certain variables, and learning *wide clauses*, i.e., clauses with a large number of literals. In this paper we present a case study to show that by building on positive results on acyclic conjunctive query evaluation in relational database theory, it is possible to construct ILP learning algorithms that are capable of discovering clauses of significantly greater depth and width. We give a detailed description of the class of clauses we consider, describe a greedy algorithm to work with these clauses, and show, on the popular ILP challenge problem of mutagenicity, how indeed our method can go beyond the depth and width barriers of current ILP systems.

1 Introduction

In recent years, it has been shown that methods from Inductive Logic Programming (ILP) [23,32] are powerful enough to discover new first-order knowledge from data, while employing a clausal representation language that is relatively easy for humans to understand. Despite these successes, it is generally acknowledged that there are issues that present fundamental challenges for the current generation of systems. Among these, two problems are particularly prominent: learning *deep clauses*, i.e., clauses where a long chain of literals is needed to reach certain variables, and learning *wide clauses*, i.e., clauses with a large number of interconnected literals.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 100–112, 2001.

In current ILP systems, these challenges are reflected in system parameters that bound the depth and width of the clauses, respectively. Practical experience in applications shows that tractable runtimes are achieved only when setting the values of these parameters to small values; in fact it is not uncommon to limit the depth of clauses to two or three, and their width to four or five. In a recent study, Giordana and Saitta [10] have shown, based on empirical simulations, that indeed there seems to be a fundamental limit for current ILP systems, and that this limit might in large parts be due to the extreme growth of matching costs, i.e., the cost of determining if a clause covers a given example. Thus, if matching costs could be reduced, it should be possible to learn clauses of significantly greater depth and width than currently achievable.

In this paper, we present an ILP algorithm and a case study which provide evidence that indeed this seems to be the case. In our algorithm, we build on positive complexity results on conjunctive query evaluation in the area of relational database theory, and employ the class of acyclic conjunctive queries where the matching problem is known to be tractable. In the domain of mutagenicity, we show that using our algorithm it is indeed possible to discover structural relationships that must be expressed in clauses that have significantly greater depth and width than those currently learnable. In fact, the additional predictive power gained by these deep and wide structures has allowed us to reach a predictive accuracy comparable to the one attained in previous studies, *without* using the additional numerical information available in these experiments.

The paper is organized as follows. In Section 2, we first briefly introduce the learning problem that is usually considered in ILP. In Section 3, we present a more detailed introduction into the matching problem and discuss the state of the art in related work on the issue. In Section 4, we then formally define the class of acyclic clauses that is used in this work, and describe its properties. Section 5 discusses our greedy algorithm which uses this class of clauses to perform ILP learning. Section 6 contains our case study in the domain of mutagenesis, and Section 7 concludes.

2 The ILP Learning Problem

The ILP learning problem is often simply defined as follows (see, e.g., [32]).

Definition 1 (ILP prediction learning problem). *Given*

- *a vocabulary consisting of finite sets of function and predicate symbols,*
- *a background knowledge language L_B, an example language L_E, and an hypothesis language L_H, all over the given vocabulary,*
- *background knowledge B expressed in L_B, and*
- *sets E^+ and E^- of positive and negative examples expressed in L_E*

such that B is consistent with E^+ and E^- ($B \cup E^+ \cup E^- \not\models \square$), find a learning hypothesis $H \in L_H$ such that

(i) H is complete, i.e., together with B entails the positive examples ($H \cup B \models E^+$)
(ii) and H is correct, i.e., is consistent with the negative examples ($H \cup B \cup E^+ \cup E^- \not\models \square$).

This problem is called the prediction learning problem because the learning hypothesis H must be such that together with B it correctly predicts (derives, covers) the positive examples, and does not predict (derive, cover) the negation of any negative example as true (otherwise the hypothesis would be inconsistent with the negative examples). For instance, if $flies(tweety)$ is a positive example and $\neg flies(bello)$ a negative one then $flies(bello)$ must not be predicted[1].

In order to decide conditions (i) and (ii) in the above definition, one has to decide for a single $e \in E^+ \cup E^-$ whether $H \cup B \models e$. This decision problem is called the *matching* or *membership* problem. We note that in the general problem setting defined above, the membership problem is *not* decidable. Therefore, in most of the cases implication is replaced by *clause subsumption* defined as follows. Let C_1 and C_2 be first-order clauses. We say that C_1 *subsumes* C_2, denoted by $C_1 \leq C_2$, if there is a substitution θ (a mapping of C_1's variables to C_2's terms) such that $C_1\theta \subseteq C_2$ (for more details see, e.g., [25]).

3 The Matching Problem: State of the Art

One of the reasons why the width and depth of the clauses in the hypothesis language are usually bounded by a small constant is that even in the strongly restricted ILP problem settings the membership problem is still NP-complete. For instance, consider the ILP prediction learning problem, where (non-constant) function symbols are not allowed in the vocabulary, the background knowledge is an extensional database (i.e., it consists of ground atoms), examples are ground atoms, and the hypothesis language is a subset of the set of definite non-recursive first-order Horn clauses, or in other words, it is a subset of the set of *conjunctive queries* [1,30]. This is one of the problem settings most frequently considered in ILP real-word applications. Although in this setting, the membership problem, i.e., the problem of deciding whether a conjunctive query implies a ground atom with respect to an extensional database, and implication between conjunctive queries are both decidable, they are still NP-complete [6]. In the ILP community, both of these problems are viewed as instances of the clause subsumption problem because implication is equivalent to clause subsumption in the problem setting considered (see, e.g., [11]). These decision problems play a central role e.g. in top-down ILP approaches (see, e.g., [25] for an overview), where the algorithm starts with an overly general clause, for instance with the empty clause, and specializes it step by step until a clause is found that satisfies the requirements defined by the user.

[1] Strictly speaking the above setting only refers to the *training error* of a hypothesis while ILP systems actually seek to minimize the true error on future examples.

As mentioned above, subsumption between first-order clauses is one of the most important operators used in different ILP methods. Since the clause subsumption problem is known to be NP-complete, different approaches can be found in the corresponding literature that try to solve it in polynomial time. Among these, we refer to the technique of identifying tractable subclasses of first-order clauses (see, e.g., [12,18,26]), to the earlier mentioned phase transitions in matching [10], and to stochastic matching [27].

In general, clause subsumption problem can be considered as a *homomorphism* problem between the relational structures that correspond to the clauses, as one has to find a function between the universes of the structures that preserves the relations (see, e.g., [16]). Homomorphisms between relational structures appear in the query evaluation problems in relational database theory or in the constraint-satisfaction problem in artificial intelligence (see, e.g., [19]). In particular, from the point of view of computational complexity, the query evaluation problem for the above mentioned class of conjunctive queries is well-studied. Research in this field goes back to the seminal paper by Chandra and Merlin [6] in the late seventies, who showed that the problem of evaluating a conjunctive query with respect to a relational database is NP-complete. In [33], Yannakakis has shown that query evaluation becomes computationally tractable if the set of literals in the query forms an *acyclic hypergraph*. This class of conjunctive queries is called *acyclic conjunctive queries*. In [13], Gottlob, Leone, and Scarcello have shown that acyclic conjunctive queries are LOGCFL-complete. The relevance of this result, besides providing the precise complexity of acyclic conjunctive query evaluation, is that acyclic conjunctive query evaluation is highly parallelizable due to the nature of LOGCFL. The positive complexity result of Yannakakis was then extended by Chekuri and Rajaraman [7] to cyclic queries of bounded query-width.

Despite the fact that the class of conjunctive queries is one of the most frequently considered hypothesis language in ILP, and that acyclic conjunctive queries form a practically relevant class of database queries, to our knowledge only the recent paper [15] by Hirata has so far been concerned with acyclic conjunctive queries from the point of view of learnability[2]. In that paper, Hirata has shown that, under widely believed complexity assumptions, a single acyclic conjunctive query is *not* polynomially predictable, and hence, it is not polynomially PAC-learnable [31]. This means that even though the membership problem for acyclic clauses is decidable in polynomial time, under worst-case assumptions the problem of *learning* these clauses is hard, so that practical learning algorithms, such as the one presented in section 5, must resort to heuristic methods.

[2] The notion of *acyclicity* appears in the literature of ILP (see, e.g., [2]), but is different from the one considered in this paper.

4 Acyclic Conjunctive Queries

In this section we give the necessary notions related to acyclic conjunctive queries considered in this work. For a detailed introduction to acyclic conjunctive queries the reader is referred to e.g. [1,30] or to the long version of [13].

For the rest of this paper, we assume that the vocabulary in Definition 1 consists of a set of constant symbols, a distinguished predicate symbol called the target predicate, and a set of predicates called the background predicates. Thus, (non-constant) function symbols are not included in the vocabulary. Examples are ground atoms of the target predicate, and the background knowledge is an extensional database consisting of ground atoms of the background predicates. Furthermore, we assume that hypotheses in L_H are definite non-recursive first-order clauses, or in the terminology of relational database theory, conjunctive queries of the form

$$L_0 \leftarrow L_1, \ldots, L_l$$

where L_0 is a target atom, and L_i is a background atom for $i = 1, \ldots, l$. In what follows, by Boolean conjunctive queries we mean first-order goal clauses of the form

$$\leftarrow L_1, \ldots, L_l$$

where the L_i's are all background atoms.

In order to define a special class of conjunctive queries, called *acyclic* conjunctive queries, we first need the notion of acyclic hypergraphs. A *hypergraph* (or *set-system*) $H = (V, E)$ consists of a finite set V called *vertices*, and a family E of subsets of V called *hyperedges*. A hypergraph is α-acyclic [9], or simply *acyclic*, if one can remove all of its vertices and edges by deleting repeatedly either a hyperedge that is empty or is contained by another hyperedge, or a vertex contained by at most one hyperedge [14,34]. Note that acyclicity as defined here is not a *hereditary* property, in contrast to e.g. the standard notion of acyclicity in ordinary undirected graphs, as it may happen that an acyclic hypergraph has a cyclic subhypergraph. For example, consider the hypergraph $H = (\{a, b, c\}, \{e_1, e_2, e_3, e_4\})$ with $e_1 = \{a, b\}$, $e_2 = \{b, c\}$, $e_3 = \{a, c\}$, and $e_4 = \{a, b, c\}$. This is an acyclic hypergraph, as one can remove step by step first the hyperedges e_1, e_2, e_3 (as they are subsets of e_4), then the three vertices, and finally, the empty hypergraph is obtained by removing the empty hyperedge that remained from e_4. On the other hand, the hypergraph $H' = (\{a, b, c\}, \{e_1, e_2, e_3\})$, which is a subhypergraph of H, is cyclic, as there is no vertex or edge that could be deleted by the above definition. In [9], other degrees of acyclicity are also considered, and it is shown that among them, α-acyclic hypergraphs form the largest class properly containing the other classes.

Using the above notion of acyclicity, now we are ready to define the class of acyclic conjunctive queries. Let Q be a conjunctive query and L be a literal of Q. We denote by $\mathrm{Var}(Q)$ (resp. $\mathrm{Var}(L)$) the set of variables occurring in Q (resp. L). We say that Q is acyclic if the hypergraph $H(Q) = (V, E)$ with $V = \mathrm{Var}(Q)$ and $E = \{\mathrm{Var}(L) : L \text{ is a literal in } Q\}$ is acyclic. For instance, from the conjunctive

queries

$$P(X,Y,X) \leftarrow R(X,Y), R(Y,Z), R(Z,X)$$
$$P(X,Y,Z) \leftarrow R(X,Y), R(Y,Z), R(Z,X)$$

the first one is cyclic, while the second one is acyclic.

In [3] it is shown that the class of acyclic conjunctive queries is identical to the class of conjunctive queries that can be represented by *join forests* [4]. Given a conjunctive query Q, the join forest $JF(Q)$ representing Q is an ordinary undirected forest such that its vertices are the set of literals of Q, and for each variable $x \in \mathrm{Var}(Q)$ it holds that the subgraph of $JF(Q)$ consisting of the vertices that contain x is connected (i.e., it is a tree).

Now we show how to use join forests for efficient acyclic query evaluation. Let E be a set of ground target atoms and B be the background knowledge as defined at the beginning of this section, and let Q be an acyclic conjunctive query with join forest $JF(Q)$. In order to find the subset $E' \subseteq E$ implied by Q with respect to B, we can apply the following method. Let $T_0, T_1, \ldots, T_k$ $(k \geq 0)$ denote the set of connected components of $JF(Q)$, where T_0 denotes the tree containing the head of Q, and let $Q_i \subseteq Q$ denote the query represented by T_i for $i = 0, \ldots, k$. The definition of the Q_i's implies that they form a partition of the set of literals of Q such that literals belonging to different blocks do not share common variables. Therefore, the subqueries $Q_0, \ldots, Q_k$ can be evaluated separately; if there is an i, $1 \leq i \leq k$, such that the Boolean conjunctive query is false with respect to B then Q implies *none* of the elements of E with respect to B, otherwise Q and Q_0 imply the same subset of E with respect to B. By definition, Q_0 implies an atom $e \in E$ if there is a substitution mapping the head of Q_0 to e and the atoms in its body into B, and Q_i $(1 \leq i \leq k)$ is true with respect to B if there is a substitution mapping Q_i's atom into B. That is, using algorithm EVALUATE given below, Q implies E' with respect to B if and only if

$$(E' \subseteq \textsc{Evaluate}(B \cup E, T_0)) \wedge \left(\bigwedge_{i=1}^{k} (\textsc{Evaluate}(B, T_i) \neq \emptyset) \right) \ .$$

It remains to discuss the problem of how to compute a join forest for an acyclic conjunctive query. Using maximal weight spanning forests of ordinary graphs, in [4] Bernstein and Goodman give the following method to this problem. Let Q be an acyclic conjunctive query, and let $G(Q) = (V, E, w)$ be a weighted graph with vertex set $V = \{L : L \text{ is a literal of } Q\}$, edge set $E = \{(u, v) : \mathrm{Var}(u) \cap \mathrm{Var}(v) \neq \emptyset\}$, and with weight function $w : E \to \mathbb{N}$ defined by

$$w : (u, v) \mapsto |\mathrm{Var}(u) \cap \mathrm{Var}(v)| \ .$$

Let $MSF(Q)$ be a maximal weight spanning forest of $G(Q)$. Note that maximal weight spanning forests can be computed in polynomial time (see, e.g., [8]). It holds that if Q is acyclic then $MSF(Q)$ is a joint forest representing Q. In addition, given a maximal weight spanning forest $MSF(Q)$ of a conjunctive query

algorithm EVALUATE

```
input: extensional database D and join tree T with root
       labeled by n_0
output: {n_0θ: θ is a substitution mapping the nodes of T into D}

let R = {n_0θ: θ is a substitution mapping n_0 into D}
let the children of n_0 be labeled by n_1, ..., n_k (k ≥ 0)
for i = 1 to k
  S = evaluate(D, T_i)      // T_i is the subtree of T rooted at n_i
  R = the natural semijoin of R and S wrt. n_0 and n_i
endfor
return R
```

Q, instead of using the method given in the definition of acyclic hypergraphs, in order to decide whether Q is acyclic, one can check whether the equation

$$\sum_{(u,v)\in MSF(Q)} w(u,v) = \sum_{x\in \mathrm{Var}(Q)} (\mathrm{Class}(x) - 1) \tag{1}$$

holds, where $\mathrm{Class}(x)$ denotes the number of literals in Q that contain x (see also [4]).[3]

5 A Greedy Algorithm

The goal of our learning algorithm is to discover sets of acyclic clauses that together are correct and complete. From the results of [16] on learning multiple clauses it follows that this problem is NP-hard, so we resort to a greedy sequential covering algorithm (see, e.g., [21]) as it is commonplace in ILP. Our sequential covering algorithm takes as input the background knowledge B and the set E of examples, calls the subroutine SINGLECLAUSE for finding an acyclic conjunctive query Q, then updates E by removing the positive examples implied by Q with respect to B, and starts the process again until no new rule is found by the subroutine. It finally prints as output the set of acyclic conjunctive queries discovered.

Now we turn to the problem of how to find a single acyclic conjunctive query[4]. In order to give the details on the subroutine SINGLECLAUSE called by

[3] The reason why $\mathrm{Class}(x) - 1$ is used in (1) is that the number of edges in a tree is equal to its number of vertices minus 1.

[4] We note that the general problem of finding a *single* consistent and complete (not necessarily acyclic) conjunctive query is a PSPACE-hard problem [17] and it is an open problem whether it belongs to PSPACE (see also [16]). On the other hand, it is not known whether it remains PSPACE-hard for the class of acyclic conjunctive queries considered in this work, or to the other three classes corresponding to β, γ, and Berge-acyclicity discussed in [9].

the algorithm, we first need the notion of *refinement operators* (see Chapter 17 in [25] for an overview). We recall that a special ILP problem setting defined at the beginning of the previous section is considered. Fix the vocabulary and let L denote the set of acyclic conjunctive queries over the vocabulary. A *downward refinement operator* is a function $\rho : L \to 2^L$ such that $Q_1 \leq Q_2$ for every $Q_1 \in L$ and $Q_2 \in \rho(Q_1)$.

algorithm SINGLECLAUSE

```
input: background knowledge B and set E = E+ ∪ E− of examples
output: either ∅ or an acyclic conjunctive query BEST satisfying
   |COVERS(BEST, B, E+)|/|E+| ≥ Pcov and ACCURACY(BEST, B, E) ≥ Pacc

BEAM = {P(x1, ..., xn) ←}        // P denotes the target predicate
BEST = ∅
LASTCHANGE = 0
repeat
  NEWBEAM = ∅
  forall C ∈ BEAM
    forall C′ ∈ ρ(C)
      if |COVERS(C′, B, E+)|/|E+| ≥ Pcov then
        if ACCURACY(C′, B, E) ≥ max(Pacc, ACCURACY(BEST, B, E)) then
          BEST = C′
          LASTCHANGE = 0
        endif
        update NEWBEAM by C′
      endif
    endfor
  endfor
  LASTCHANGE = LASTCHANGE + 1
  BEAM = NEWBEAM
until BEAM = ∅ or LASTCHANGE > Pchange
return BEST
```

Algorithm SINGLECLAUSE applies beam search for finding a single acyclic conjunctive query. Its input is B and the current set E of examples. It returns the acyclic conjunctive query BEST that covers a sufficiently large (defined by P_{cov}) part of the positive examples and has accuracy at least P_{acc}, where P_{cov} and P_{acc} are user defined parameters. If it has not found such an acyclic conjunctive query, then it returns the empty set. In each iteration of the outer (repeat) loop, the algorithm computes the refinements for each acyclic conjunctive query in the beam stack, and if a refinement is found that is better than the best one discovered so far then it will be the new best candidate. The beam stack is updated according to the rules' quality measured by ACCURACY. Finally we note that the outer loop is terminated if no candidate refinement has been generated

or in the last P_{change} iterations of the outer loop the best rule has not been changed, where P_{change} is a user defined parameter.

6 Case Study: Mutagenesis

Chemical mutagens are natural or artificial compounds that are capable of causing permanent transmissible changes in DNA. Such changes or *mutations* may involve small gene segments as well as whole chromosomes. *Carcinogenic* compounds are chemical mutagens that alter the DNA's structure or sequence harmfully causing cancer in mammals. A huge amount of research in the field of organic chemistry has been focusing on identifying carcinogen chemical compounds.

The first study on using ILP for predicting mutagenicity in nitroaromatic compounds along with providing a Prolog database was published in [29]. This database consists of two sets of nitroaromatic compounds from which we have used the regression friendly one containing 188 compounds. Depending on the value of log mutagenicity, the compounds were split into two disjoint sets (active consisting of 125 and inactive consisting of 63 compounds). The basic structure of the compounds is represented by the background predicates 'atm' and 'bond' of the form

atm(Compound_Id,Atom_Id,Element,Type,Charge),

bond(Compound_Id,Atom1_Id,Atom2_Id,BondType) ,

respectively. Thus, the background knowledge B can be considered as a labeled directed graph. In order to work with *undirected graph*, for each fact $\text{bond}(c, u, v, t)$ we have added a corresponding fact $\text{bond}(c, v, u, t)$ to B. In addition, in our experiments we have also included the background predicates

- benzene, carbon_6_ring, hetero_aromatic_6_ring, ring6,
- carbon_5_aromatic_ring, carbon_5_ring, hetero_aromatic_5_ring, ring5,
- nitro, and methyl.

These predicates define building blocks for complex chemical patterns (for their definitions see the Appendix of [29]). We note that we have not used the available numeric information (i.e., charge of atoms, log P, and ϵ_{LUMO}).

In our experiments we used a simple refinement operator allowing only adding new literals to the body of an acyclic conjunctive query, and not allowing the usual operators such as unification of two variables or specialization of a variable. That is, a refinement of an acyclic conjunctive query is obtained by selecting one of its literals, and, depending on the predicate symbol of the selected literal, by adding a set of literals to its body as follows. If the literal is the head of the clause we add either a single 'atm' literal or a set of literals corresponding to one of the building blocks. If the literal selected is an 'atm' then we add either a new atom connected by a bond fact with the selected one, or we add a building block containing the selected atom. If a bond literal has been selected then we add a building block containing the current bond. Such building blocks are a common

element specifiable in several declarative bias languages already in use in ILP (see e.g. the relational clichés of FOCL [28] or the lookahead specifications of TILDE [5]); at present, they are simply given as part of the refinement operator[5].

As an example, let

$$Q : \text{active}(x_1, x_2) \leftarrow \ldots, L, \ldots$$

be an acyclic conjunctive query, where $L = \text{bond}(x_1, x_i, x_j, 7)$. Then a refinement of Q with respect to L and building block benzene is the acyclic conjunctive query

$$\begin{aligned} Q' = Q \cup \{&\text{bond}(x_1, x_j, y_1, 7), \text{bond}(x_1, y_1, y_2, 7), \text{bond}(x_1, y_2, y_3, 7), \\ &\text{bond}(x_1, y_3, y_4, 7), \text{bond}(x_1, y_4, x_i, 7), \text{atm}(x_1, y_1, c, u_1, v_1), \\ &\text{atm}(x_1, y_2, c, u_2, v_2), \text{atm}(x_1, y_3, c, u_3, v_3), \text{atm}(x_1, y_4, c, u_4, v_4), \\ &\text{benzene}(x_1, x_i, x_j, y_1, y_2, y_3, y_4)\} \ , \end{aligned}$$

where the y's, u's, and v's are all new variables. Note that despite the fact that the new bond literals together with L form a cycle of length 6, Q' is acyclic, as we have also attached the benzene literal containing the six corresponding variables. It holds in general that the refinement operator used in our work does not violate the acyclicity property. Finally we note that only properly subsumed refinements have been considered (i.e., if Q' is a refinement of Q then $Q' \not\leq Q$).

In order to see how our restriction on the hypothesis language influences the predictive accuracy, we have used 10-fold cross-validation with the 10 partitions given in [29]. Setting parameters P_{cov} to 0.1, P_{acc} to 125/188 (default accuracy), the size of the beam stack to 100, and P_{change} to 3 (note that this is *not* a depth bound), we obtained 87% accuracy. Using the ILP system Progol [22], the authors of [29] report 88% accuracy, and a similar result, 89% was achieved by STILL [27] on the same ten partitions. However, in contrast to our experiment, in the Progol and STILL experiments, the numeric information was considered as well.

As an example, one of the rules discovered independently in each of the ten runs is

$$\begin{aligned} &\text{active}(x_1, x_2) \leftarrow \\ &\quad \text{atm}(x_1, x_3, c, 27, x_4), \\ &\quad \text{bond}(x_1, x_3, x_5, x_{25}), \text{bond}(x_1, x_5, x_6, x_{26}), \text{bond}(x_1, x_6, x_7, x_{27}), \\ &\quad \text{bond}(x_1, x_7, x_8, x_{28}), \text{bond}(x_1, x_8, x_9, x_{29}), \text{bond}(x_1, x_9, x_3, x_{30}), \\ &\quad \text{atm}(x_1, x_5, x_{10}, x_{11}, x_{12}), \text{atm}(x_1, x_6, x_{13}, x_{14}, x_{15}), \text{atm}(x_1, x_7, x_{16}, x_{17}, x_{18}), \\ &\quad \text{atm}(x_1, x_8, x_{19}, x_{20}, x_{21}), \text{atm}(x_1, x_9, x_{22}, x_{23}, x_{24}), \\ &\quad \text{ring6}(x_1, x_3, x_5, x_6, x_7, x_8, x_9), \\ &\quad \text{bond}(x_1, x_7, x_{31}, x_{32}), \text{atm}(x_1, x_{31}, c, 27, x_{33}) \end{aligned} \tag{2}$$

[5] Note that the use of such building blocks facilitates the search by making wide and deep clauses reachable in fewer steps, but of course does not change the complexity of the membership problem. Thus, even when given these building blocks, such clauses would be difficult to learn for other ILP learners due to the intractable cost of matching.

(see also Fig. 1). Applying the notion of variable depth given in [25], the depth of the above rule is 7 according to the depth of the deepest variable x_{22}. Furthermore, its width is 15. Finally we note that using the standard Prolog backtracking technique, just evaluating the single rule above would take on the order of hours.

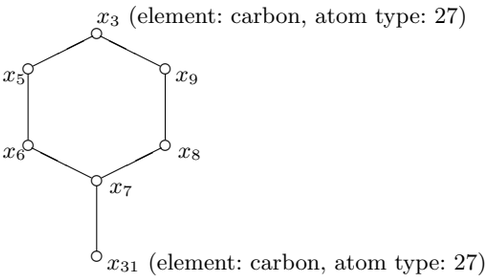


Fig. 1. A graphical representation of the body of rule (2).

7 Conclusion

In this paper, we have taken the first steps towards discovery of deep and wide first-order structures in ILP. Taking up the argument recently put forward by [10], our approach centrally focuses on the matching costs caused by deep and wide clauses. To this end, from relational database theory [1,30], we have introduced a new class of clauses, *α-acyclic conjunctive queries*, which has not previously been used in practical ILP algorithms. Using the algorithms summarized in this paper, the matching problem for acyclic clauses can be solved efficiently. As shown in our case study in the domain of mutagenicity, with an appropriate greedy learner as presented in the paper, it is then possible to learn clauses of significantly greater width and depth than previously feasible, and the additional predictive power gained by these deep and wide structures has in fact allowed us to reach a predictive accuracy comparable to the one attained in previous studies, *without* using the addition numerical information available in these experiments.

Based on these encouraging preliminary results, further work is necessary to substantiate the evidence presented in this paper. Firstly, in the case study presented here, we have used quite a simple greedy algorithm, so that further improvements seen possible with more sophisticated search strategies (see e.g. [20]). Secondly, further experiments are of course necessary to examine in which type of problem the advantages shown here will also materialize; we expect this to be the case in all problems involving structurally complex objects or relationships. To facilitate the experiments, we will switch to a refinement operator based on a declarative bias language (see [24] for an overview), as is commonplace in ILP. Finally, it appears possible to generalize our results to an even

larger class of clauses, by considering certain classes of cyclic conjunctive queries which are also solvable in polynomial time (see e.g. [7]).

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, Reading, Mass., 1995.
2. H. Arimura. Learning acyclic first-order Horn sentences from entailment. In M. Li and A. Maruoka, editors, *Proceedings of the 8th International Workshop on Algorithmic Learning Theory*, volume 1316 of *LNAI*, pages 432–445, Springer, Berlin, 1997.
3. C. Beeri, R. Fagin, D. Maier, and M. Yannakakis. On the desirability of acyclic database schemes. *Journal of the ACM*, 30(3):479–513, 1983.
4. P. A. Bernstein and N. Goodman. The power of natural semijoins. *SIAM Journal on Computing*, 10(4):751–771, 1981.
5. H. Blockeel and L. D. Raedt. Lookahead and discretization in ILP. In N. Lavrač and S. Džeroski, editors, *Proceedings of the 7th International Workshop on Inductive Logic Programming*, volume 1297 of *LNAI*, pages 77–84, Springer, Berlin, 1997.
6. A. K. Chandra and P. M. Merlin. Optimal implementations of conjunctive queries in relational databases. In *Proceedings of the 9th ACM Symposium on Theory of Computing*, pages 77–90. ACM Press, 1977.
7. C. Chekuri and A. Rajaraman. Conjunctive query containment revisited. *Theoretical Computer Science*, 239(2):211–229, 2000.
8. T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, Mass., 1990.
9. R. Fagin. Degrees of acyclicity for hypergraphs and relational database schemes. *Journal of the ACM*, 30(3):514–550, 1983.
10. A. Giordana and L. Saitta. Phase transitions in relational learning. *Machine Learning*, 41(2):217–251, 2000.
11. G. Gottlob. Subsumption and implication. *Information Processing Letters*, 24(2):109–111, 1987.
12. G. Gottlob and A. Leitsch. On the efficiency of subsumtion algorithms. *Journal of the ACM*, 32(2):280–295, 1985.
13. G. Gottlob, N. Leone, and F. Scarcello. The complexity of acyclic conjunctive queries. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 706–715. IEEE Computer Society Press, 1998.
14. M. Graham. On the universal relation. Technical report, Univ. of Toronto, Toronto, Canada, 1979.
15. K. Hirata. On the hardness of learning acyclic conjunctive queries. In *Proceedings of the 11th International Conference on Algorithmic Learning Theory*, volume 1968 of *LNAI*, pages 238–251. Springer, Berlin, 2000.
16. T. Horváth and G. Turán. Learning logic programs with structured background knowledge. *Artificial Intelligence*, 128(1-2):31–97, 2001.
17. J.-U. Kietz. Some lower bounds for the computational complexity of inductive logic programming. In P. Brazdil, editor, *Proceedings of the European Conference on Machine Learning*, volume 667 of *LNAI*, pages 115–123. Springer, Berlin, 1993.
18. J.-U. Kietz and M. Lübbe. An efficient subsumption algorithm for inductive logic programming. In W. Cohen and H. Hirsh, editors, *Proc. Eleventh International Conference on Machine Learning (ML-94)*, pages 130–138, 1994.

19. Kolaitis and Vardi. Conjunctive-query containment and constraint satisfaction. *JCSS: Journal of Computer and System Sciences*, 61(2):302–332, 2000.
20. N. Lavrač and S. Džeroski. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, 1994.
21. T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
22. S. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13(3-4):245–286, 1995.
23. S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19/20:629–680, 1994.
24. C. Nédellec, C. Rouveirol, H. Adé, F. Bergadano, and B. Tausend. Declarative bias in ILP. In L. De Raedt, editor, *Advances in Inductive Logic Programming*, pages 82–103. IOS Press, 1996.
25. S.-H. Nienhuys-Cheng and R. Wolf. *Foundations of Inductive Logic Programming*, volume 1228 of *LNAI*. Springer, Berlin, 1997.
26. T. Scheffer, R. Herbrich, and F. Wysotzki. Efficient Θ-subsumption based on graph algorithms. In S. Muggleton, editor, *Proceedings of the 6th International Workshop on Inductive Logic Programming*, volume 1314 of *LNAI*, pages 212–228, Springer, Berlin, 1997.
27. M. Sebag and C. Rouveirol. Resource-bounded relational reasoning: Induction and deduction through stochastic matching. *Machine Learning*, 38(1/2):41–62, 2000.
28. G. Silverstein and M. Pazzani. Relational cliches: Constraining constructive induction during relational learning. In Birnbaum and Collins, editors, *Proceedings of the 8th International Workshop on Machine Learning*, pages 203–207, Morgan Kaufmann, San Mateo, CA, 1991.
29. A. Srinivasan, S. Muggleton, M. J. E. Sternberg, and R. D. King. Theories for mutagenicity: A study in first-order and feature-based induction. *Artificial Intelligence*, 85(1/2), 1996.
30. J. D. Ullman. *Database and Knowledge-Base Systems, Volumes I and II*. Computer Science Press, 1989.
31. L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1985.
32. S. Wrobel. Inductive logic programming. In G. Brewka, editor, *Advances in Knowledge Representation and Reasoning*, pages 153–189. CSLI-Publishers, Stanford, CA, USA, 1996. Studies in Logic, Language and Information.
33. M. Yannakakis. Algorithms for acyclic database schemes. In *Proceedings of the 7th Conference on Very Large Databases, Morgan Kaufman pubs. (Los Altos CA), Zaniolo and Delobel(eds)*, 1981.
34. C. T. Yu and Z. M. Ozsoyoglu. On determining tree query membership of a distributed query. *INFOR*, 22(3), 1984.

Eliminating Useless Parts in Semi-structured Documents Using Alternation Counts

Daisuke Ikeda[1], Yasuhiro Yamada[2], and Sachio Hirokawa[1]

[1] Computing and Communications Center,
Kyushu University, Fukuoka 812-8581, Japan
{daisuke,hirokawa}@cc.kyushu-u.ac.jp
[2] Graduate School of Information Science and Electrical Engineering,
Kyushu University, Fukuoka 812-8581, Japan
yshiro@matu.cc.kyushu-u.ac.jp

Abstract. We propose a preprocessing method for Web mining which, given semi-structured documents with the same structure and style, distinguishes useless parts and non-useless parts in each document without any knowledge on the documents. It is based on a simple idea that any n-gram is useless if it appears frequently. To decide an appropriate pair of length n and frequency a, we introduce a new statistic measure *alternation count*. It is the number of alternations between useless parts and non-useless parts. Given news articles written in English or Japanese with some non-articles, the algorithm eliminates frequent n-grams used for the structure and style of articles and extracts the news contents and headlines with more than 97% accuracy if articles are collected from the same site. Even if input articles are collected from different sites, the algorithm extracts contents of articles from these sites with at least 95% accuracy. Thus, the algorithm does not depend on the language, is robust for noises, and is applicable to multiple formats.

1 Introduction

Data mining is a research field to develop tools that find useful knowledge from databases [3,4,14]. In this field, databases is assumed to have explicit and static structures. On the other hand, resources on the WWW do not have such structures. Web mining is a field of mining from such resources and text mining is mining from unstructured or semi-structured documents. We consider Web or text mining from semi-structured documents. A semi-structured document have tree structures, such as HTML/XML files, BiBTeX files, etc [1].

Since resources on the Web are widely distributed and heterogeneous, it is important to collect documents and clean them. When we collect a large amount of documents, we use hyperlinks for efficiency. For example, search engines provide hyperlinks. Since a hyperlink is not created by the collecter, we can not assume that collected documents are well cleaned. Some of them are written in different languages and are far from the desired topic. Thus, Web mining algorithms and preprocessors should be robust for noise and should be applicable to any natural and markup languages.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 113–127, 2001.

Usual mining algorithms assume that collected documents are written in the same language and preprocess them with some knowledge, such as the grammar of HTML to remove tags, stop word lists [10], stemming technique [15], morpheme analysis, etc. They depends on natural and markup languages. Some algorithms require additional input documents as background knowledge. In addition to an input set of documents, the algorithm in [5] requires another set of documents in order to remove substrings highly common to both sets.

In this paper, we present an algorithm that cleans collected documents without any knowledge on them. From input documents, this algorithm finds a set of frequent n-grams. Using this set, we can eliminate tags or directives, and stereotyped expressions in the documents because they are common to the collected documents and so useless. Eliminating or finding *useless* parts contrasts with usual mining algorithms which find *useful* knowledge such as association rules [3], association patterns [4], word association patterns [5], and episode rules [14].

An input for our algorithm is a set of semi-structured documents which contain the same structure and style such as static Web pages in the same site or dynamic pages generated with search facility. Since the number of Web sites providing search facilities is increasing [7], the number of Web pages applicable to our algorithm is large. In such pages, there exist frequent substrings, such as the name of the site, navigation and advertisement links, etc. Moreover, there exist common tags or directives which structure texts because they have the same structure and style. If we see such pages, we usually put our mind on the variable part and ignore invariable parts. In this sense, substrings to describe the same structure and style in such pages are useless.

We treat a document as just a string and define any frequent n-grams are useless. An n-gram is just a string with length n, so that our algorithm does not depend on natural and markup languages. Once an appropriate pair (n, a), which is called a *cut point*, of a length n and frequency a is decided, we divide each of documents into two parts using the pair as follows: if the frequency of an n-gram is in the top a percent of the frequencies of all n-grams, then the n-gram is useless.

To decide an appropriate pair (n, a), we introduce a new statistic measure *alternation count*. It is the number of changes between useless parts and non-useless parts in a document. For a set D of documents, alternation count of D is the sum of all alternation counts. An alternation count shows how many times useless (or non-useless) parts appear in documents. A large alternation count splits a document into too small pieces. In this case, structures of each document are destroyed. Therefore, our algorithm searches cut points from $(2, 1)$ while the alternation count of the current cut point is decreasing. The algorithm stops if the alternation count becomes to be greater than the current one when it increases n or a by one.

We define an *optimal* cut point (n, a) which attains a locally minimum alternation count and develop an algorithm that finds an optimal cut point. It runs in $O(n^2N + nN \log N)$ time, where N is the total length of input and n is the

length of an optimal cut point. Experimentally, n is less than 30 and $n \ll N$, so the time complexity is approximately $O(N \log N)$.

We present experimental results using news articles as input for the algorithm. The articles are written in English or Japanese. For articles collected from the same site, the algorithm detects the tag regions of HTML files and highly common expressions in the articles as useless. It extracts the news contents as non-useless parts with more than 97% accuracy even if non-articles are contained as noise data. Therefore, the algorithm does not depend on the language and is robust for noises.

To evaluate experimental results, we manually define non-useless parts of articles for each data set. Comparing the manually defined division with the division by the algorithm, the accuracy is defined to be the number of letters categorized to the same part (useless or non-useless) to the total length of the input.

We also evaluate the accuracy for documents collected from different sites with any combination of English and Japanese sites. The algorithm extracts contents of articles from these sites with at least 95% accuracy. Since the algorithm simply counts n-gram frequencies, a extremely small set of articles would be ignored in a combination with large data sets. However, some expriments show that the algorithm detects the contents of articles in such a small set as well as those in a large set.

This paper is organized as follows. In the next section, basic notations are given and then the key notion, the alternation count, is introduced. In Section 3, we present an algorithm that divides a document of given documents into useless and non-useless parts. Complexity required by the algorithm is presented in Section 3.1. Experimental results are shown in Section 4.

2 Alternation Count and Optimal Cut Point

2.1 Preliminaries

The set Σ is a finite alphabet. Let $x = a_1 \cdots a_n$ ($a_i \in \Sigma$ for each i) be a string over Σ. We denote the *length* of x by $|x|$. An *n-gram* is a string whose length is n. For an integer $1 \leq i \leq |x|$, we denote by $x[i]$ the ith letter of x. Let x and y be two strings. The concatenation of x and y is denoted by $x \cdot y$ or simply by xy. We denote $x = y$ if $|x| = |y|$ and $x[i] = y[i]$ for each $1 \leq i \leq |x|$.

For a string x, if there exist strings $u, v, w \in \Sigma^*$ such that $x = uvw$, we say that v is a *substring* of x. An *occurrence* of v in x is a positive integer i such that $x[i] \cdots x[i + |v| - 1] = v$. Using the occurrence and the length of v, v is also denoted by $x[i..i + |v| - 1]$.

If $\Sigma = \{0, 1\}$, then a string over Σ is called a *binary* string. For $i \in \Sigma$ and $x \in \Sigma^*$, $[x]_i$ denotes the number of i's in x. For two binary strings x and y with the same length, bitwise "and" and "exclusive-or" operations defined as follows. $x\&y$ is a binary string with length $|x|$ such that $x\&y[i] = 1$ if $x[i] = y[i] = 1$ and $x\&y[i] = 0$ otherwise. $x \hat{} y$ is also a binary string with length $|x|$ such that

$x \hat{} y[i] = 1$ if $x[i] \neq y[i]$ and $x \hat{} y[i] = 0$ otherwise. For example, if $x = 01101$ and $y = 11100$, then $x\&y = 01100$, $x \hat{} y = 10001$, $[x\&y]_0 = 3$, and $[x \hat{} y]_1 = 2$.

Let x be a string (not limited to be binary) and $W = \{v_1, \ldots, v_n\}$ be a set of substrings of x. A *range string* of W on x, denoted by $r_x(W)$, is a binary string with length $|x|$ such that $r_x(W)[j] = 0$ if $i \leq j \leq i + |v_k| - 1$ for some occurrence i of v_k $(1 \leq k \leq n)$ and $r_x(W)[j] = 1$ otherwise. A successive 0's on the range string shows intervals on x covered by the substrings in W. For example, let $x = accbaacbc$ and $W = \{cb, ba\}$. Then $r_x(W) = 110001001$.

2.2 Alternation Count

In this section, we introduce the key notion *alternation count*. We consider semi-structured documents with the same structure and style such as static pages of one site and dynamic pages created by a search facility. In such documents, there exists frequent substrings for the structure and style and many users are not interested in them. Therefore, we define useless parts of documents as follows.

Definition 1. *Let D be a set of strings. Then, a substring of a string in D is said to be* useless *when it appears frequently in D.*

We treat a semi-structured document as just a string. Note that we do not define "importance", "siginificance", or "usefulness" like other researchs on text and Web mining.

Next, we consider how many times a substring appears we can say that it does "frequently". The measure for it is new notion alternation count. It is, given a string and a set of substrings of the string, the number of changes from a part of the string covered by given substrings to the other part, and vice versa.

Definition 2. *Let x be a string and W be a set of substrings of x. Then, the* alternation count *of W on x, which is denoted by $A_x(W)$, is the number of boundaries between different value's (0 and 1) on the range string $r_x(W)$.*

Example 1. Let $x = accbaacbc$ and $W = \{cb, ba\}$. Then $A_x(W) = 4$ because $x = ac\underline{cba}a\underline{cb}c$ (a part of underlined letters is cb or ba) and $r_x(W) = 110001001$.

The above definition is easily extended to a set of strings instead of a single string x. The alternation count of W on a set D of strings is the sum of alternation counts of $x \in D$ and denoted by $A_D(W)$.

2.3 Optimal Cut Point

Our algorithm is required to receive a set of semi-structured documents and divide them into two parts of substrings — useless parts and non-useless parts.

Since we treat a semi-structured document as just a string, an input for our algorithm is a set $D = \{x_1, x_2, \ldots, x_n\}$ of strings. To express useless or non-useless parts, we use a set of substrings of $x_i \in D$. Thus, our algorithm is required to receive D and decide W such that consecutive 0's on $r_{x_i}(W)$ $(i = 1, 2, \ldots, n)$ cover substrings on x_i for the structure and style.

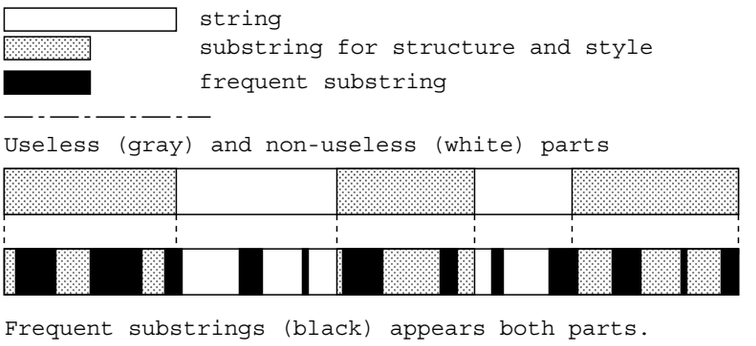


Fig. 1. Two strings are the same. The above one shows that where are substrings for the structure and style. The other one shows that the frequent substrings (black parts) fails to cover gray parts

The most simple method to find frequent substrings of D is to enumerate all substrings of D and decide a boundary between frequent substrings and non-frequent ones according to some measure. However, W constructed by this method may contain short substrings and they appears in non-useless parts as well as useless parts. And, the method requires a large time complexity because there exist $O(N^2)$ substrings for a string with length N.

Instead of this, we make the algorithm to decide the appropriate length of substring as well as the appropriate frequency. In other word, we use n-grams instead of substrings with any length. In our algorithm, the frequency is expressed by a percentage. A pair (n, a) of a length and a frequency decides W such that W is the set of the top a percent frequent n-grams in D. In the sequel, we denote $A_D(W)$ by $A_D(n, a)$. The pair (n, a) is called a *cut point* of D. An n-gram is said to be *frequent* on a cut point (n, a) if the n-gram in W decided by (n, a).

Since an appropriate cut point depends on input documents, we make the algorithm to it automatically. First, we consider what is an appropriate cut point. Two strings in Fig. 1 show the same string. The above string shows that substrings for the structure and style are colored with gray. An appropriate (n, a) covers gray parts with frequent n-grams. The below string shows that frequent n-grams appear in both gray and white parts. In this case, the alternation count is larger than the ideal alternation count and substrings for the structure and style are destroyed.

Since short substrings seems not to construct structures and styles, they appear everywhere. Therefore, an alternation count may be large if n is too small. An alternation count also may be large if a is smaller than the ideal one since increasing a connects separate frequent n-grams. If n or a is greater than the corresponding ideal one, the alternation count also become large. Thus, an

appropriate cut point if both n and a are enough large and the pair (n, a) attains a locally minimum alternation count.

A *path* is a sequence of cut points $(n_1, a_1), (n_2, a_2), \ldots, (n_k, a_k)$ such that (1) either $n_{i+1} = n_i + 1$ or $a_{i+1} = a_i + 1$ for each $i = 1, 2, \ldots, k-1$, (2) $A_D(n_i, a_i) > A_D(n_{i+1}, a_{i+1})$ for each $i = 1, 2, \ldots, k-1$, and (3) $A_D(n_k, a_k) < A_D(n_k + 1, a_k)$ and $A_D(n_k, a_k) < A_D(n_k, a_k + 1)$. A cut point (n, a) is *optimal* if there exists a path from $(2, 1)$ to (n, a). Note that, there exist some optimal cut points.

The trivial initial cut point is $(1, 1)$. However, 1-gram is too short to describe the structure and style of documents. Thus, we define the initial cut point is $(2, 1)$.

Using above notions, we define that eliminating useless parts is, given a set D of strings, to find an optimal cut point of D.

3 Algorithm

In this section, we describe `FindOptimal` that finds an optimal cut point (n, a) (see Fig. 3). From the initial cut point $(2, 1)$, the algorithm compares alternation counts on the next two cut points with the currnet one. It stops when alternation counts on both next two cut points are greater than the current one.

The algorithm does not remove any tags or directives of semi-structured documents. It only modifies documents according to the following conventional preprocessing rules: tabs and newlines are treated as a space, and consecutive spaces are compressed into one space. An input for the algorithm is a set of strings preprocessed according to the above rules.

`FindOptimal` uses two subroutines `alternation` (see Fig. 2) and `countsort`. The subroutine `countsort` receives an integer n and a set D of strings, then counts all n-grams in D and sorts them by the number of their occurrences. The subroutine keeps the n-grams and the numbers of their occurrences in a hash table. It returns an array of the counted substrings sorted in the decreasing order.

The subroutine `alternation` receives a set D of strings, an array O of strings, a length n, and a percentage a. The variable W used in `alternation` keeps the first $a/100$ strings in the sorted array O which is an output of `countsort`. Using W, the subroutine construct a range string r and then counts the boundaries, which is the alternation count on (n, a).

The main algorithm `FindOptimal` (see Fig. 3) receives a set D of strings. It counts the alternation count on the current cut point and also counts alternation counts on next two candidates, $(n, a + 1)$ and $(n + 1, a)$. After comparison alternation counts on these three cut points, it decides the next cut point. If both $A_D(n, a + 1)$ and $A_D(n + 1, a)$ are small than $A_D(n, a)$, the algorithm selects the cut point providing a smaller alternation count. If there is no next cut point providing a smaller alternation count than the current one, then it returns the currnet cut point as an optimal cut point.

```
function alternation (var D: set of strings;
  O: array of strings, n: integer ; a: integer): integer;
  var
    i: integer;
    s, x: string;
    W: hash table;
    r: array [1..|x|] of integers;
  begin
    x := string concatenated all strings in D;
    for i := 1 to |x| do r[i] := 1;
    W := substrings
      from the first substring to the a/100-th substring in O;
    for i := 1 to |x| do begin
      s := x[i..i + n] ;
      if s ∈ W then
        r[i..i + n] := 0 ;
    end {for}
    count boundaries between r[i] = 0 and r[i] = 1;
    return the boundaries;
  end ;
```

Fig. 2. The subroutine returns the alternation count of D on (n, a)

3.1 Complexity

In this section, we discuss the time complexty required by `FindOptimal`. Let D be an input set of strings and N be the total length of D.

First, we estimate the complexity required by the subroutine `countsort`. It is required $O(1)$ time to add a new n-gram to a hash table or to check if a given n-gram is already stored in the hash table. Since the subroutine counts all substrings with the same length, there exist at most $O(N)$ n-grams. Therefore, `countsort` needs $O(N)$ time to construct the hash table and $O(N \log N)$ time to sort n-grams. Thus, `countsort` runs in $O(N \log N)$ time.

Next, we consider the subroutine `alternation`. Let (D, O, n, a) be an input for the subroutine. $O(N)$ time is required to construct a hash table W. In the last **for** loop, check if $s \in W$ requires $O(1)$ using the hash table W and writing 0 on $r[i..i + n]$ requires $O(n)$ time. Therefore, this loop is completed in $O(nN)$ time. Counting boundaries is done in $O(N)$ by scanning r from left to right. Thus, the subroutine runs in $O(nN)$ time.

Finally, we estimate the time complexity of `FindOptimal`. Let (n_f, a_f) be the final output of `FindOptimal`. `FindOptimal` calls `countsort` $n_f - 1$ times and `alternation` at most $3(n_f + a_f - 2) + 1$ times because it passes through $n_f + a_f - 2$ cut points from the initial cut point $(2, 1)$ to (n_f, a_f). Thus, the routine runs in the following time:

$$\begin{aligned} & O(n_f N \log N + (n_f + a_f) n_f N) \\ = {} & O(n_f N \log N + n_f^2 N + a_f n_f N) \\ = {} & O(n_f^2 N + n_f N \log N), \end{aligned}$$

```
procedure FindOptimal (var D: set of strings);
  {FindOptimal finds an optimal cut point.}
  var n, a: integer;
  {n and a keep the current length and percentage, respectively.}
  var val_0, val_1, val_2: integer;
  var O_cur, O_next: array of strings;
  begin
    n := 2; a := 1 ; {initialize n and a.}
    O_cur := countsort(D, n);
    val_0 :=alternation(D, O_cur, n, a) ;
    {val_0 keeps the alternation count on the current (n, a).}
    while (n ≤ max{|d| | d ∈ D}) and (a < 100) do begin
      if countsort(D, n + 1) is not done then
         O_next := countsort(D, n + 1);
      val_1 :=alternation(D, O_cur, n, a + 1);
      val_2 :=alternation(D, O_next, n + 1, a);
      {val_1 and val_2 keep the alternation counts on next candidates.}
      if (val_0 ≤ val_1) and (val_0 ≤ val_2) then
        goto OUTPUT ;
      else if (val_0 > val_1) and ( val_1 < val_2) then begin
        a := a + 1 ;
        val_0 := val_1 ;
      end
      else if ((val_0 > val_2) and (val_2 ≤ val_1)) then begin
        n := n + 1 ;
        val_0 := val_2 ;
        O_cur := O_next ;
      end
    end ; {while}
    OUTPUT:
    report (n, a) ;
  end ;
```

Fig. 3. The main algorithm `FindOptimal` outputs an optimal cut point using two subroutines `countsort` and `alternation`

where a_f is a non-negative constant less than 100. Our experimental results show that n_f is less than 30 and $n \ll N$ (see Section 4). Thus, the time complexity is approximately $O(N \log N)$.

4 Experiments

We use news articles as input for `FindOptimal`. An article we use is written in English or Japanese. It is provided as an HTML file which has the headline and the body of it.

We have two types of experiments depending on the the number of sites from which we collect articles: articles from the same site (see Section 4.2) and articles from different sites (see Section 4.3).

4.1 Evaluation

To evaluate an outputed cut point, we utilize two approaches. We modify HTML files in which any frequent n-gram on the cut point is colored with gray like *ac*cba*a*cb*c*. A colored letter corresponds to 0 on the range string 110001001.

The other approache is to calculate accuracy, recall, and precision using a binary string called a *correct string*. We can conclude that `FindOptimal` outputs an appropriate cut point if a range string $r_x(n, a)$ is similar to the corresponding correct string.

Let D be an input for `FindOptimal` and x be a string concatenated all strings in D. A correct string c is a binary string with length $|x|$ such that, for $1 \leq i \leq |c|$, $c[i] \in \{0, 1\}$ is decided according to manually specified pairs of delimiters. Let (l, r) be a pair of delimiters and u be a substring of c. Then, $u = 11 \cdots 1$ if lur be a substring of x, $u = 00 \cdots 0$ otherwise. A substring surrounding with the pair is the headline or the body of an article in our experiments. The positions corresponding to the substring are filled with 1 in the correct string.

We define that the body and the headline of an article are not useless, extract manually left and right delimiters from each data set, and construct correct strings using pairs of delimiters. Then, using two binary string, the correct string c and the range string r, we define that accuracy is $[c \hat{} r]_0/|r|$, recall is $[c\&r]_1/[c]_1$, and precision is $[c\&r]_1/[r]_1$. The accuracy is the ratio of positions i such that $c[i] = r[i]$ to the total length of input documents.

4.2 Articles from the Same Site

In this section, two sets of documents are considered. One is a set of 76 articles obtained from "The Washington Post (`http://www.washingtonpost.com/`)". This set is denoted by *WPOST*. All pages linked from the URL are collected and then non-article pages are removed manually. Articles in any categories are included, and so do any other data sets in this paper. The total size of WPOST

<HTML><HEAD> <style type="text/css">□□□1370□□□ <META NAME="edition" CONTENT="M2"> <META NAME="document_name" CONTENT="A31243-2001Jan22"> <META NAME="source" CONTENT="Post"> <META NAME="section" CONTENT="DM"> <META NAME="page" CONTENT="E01 "> <META NAME="column" CONTENT=""> <META NAME="slug" CONTENT="BANK23"> <META NAME="timestamp" CONTENT="07:08 AM"> <META NAME="category" CONTENT="BIZ"> <META NAME="wordcount" CONTENT="0"> <META NAME="sourceNumber" CONTENT="6"> <!–plsfield:title–> <TITLE>McColl Shuts Books on an Era (washingtonpost.com)</TITLE> </HEAD>□□□15200□□□ <!–plsfield:headline–> <FONT FACE="Arial,Helvetica" SIZE="+1"><B>McColl Shuts Books on an Era</B></FONT> <!–plsfield:stop–>□□□2050□□□ <A HREF="/cgi-bin/gx.cgi/AppLogic+FTContentServer?pagename=wpni/email&articleid=A31243-2001Jan22&node=business"><B>E-Mail This Article</B></A>
</FONT></TD> □□□470□□□<A HREF="/ac2/wp-dyn/A31243-2001Jan22?language=printer"><B> Printer-Friendly Version</B></A>
</FONT></TD> <TD WIDTH="8" HEIGHT="1"><SPACER TYPE="block" WIDTH="8" HEIGHT="1"></TD></TR> <TR><TD COLSPAN="4" WIDTH="226" HEIGHT="8"><SPACER TYPE="block" WIDTH="226" HEIGHT="8"></TD></TR></TABLE> </TD></TR> <TR><TD WIDTH="228" HEIGHT="1"><SPACER TYPE="block" WIDTH="226" HEIGHT="1"></TD></TR></TABLE> </TD></TR></TABLE> <FONT SIZE="2"> <!–plsfield:byline–> <I>By Kathleen Day</I>
 <!–plsfield:credit–> Washington Post Staff Writer
 <!–plsfield:disp_date–> Tuesday, January 23, 2001; Page E01
 </FONT> </P> <!–plsfield:description–> <P><P></P> <P><P>The expected resignation tomorrow of Hugh McColl as head of Bank of America symbolically marks the end of an era in banking, where for two decades mergers have been an engine of growth and the idea that bigger is better has been gospel.</P> <P><P> His departure from the nation's largest consumer bank comes less than a year after his fierce crosstown competitor in Charlotte, Edward C.□□□4580□□□ "They have a mutual respect for each other," said Virginia Stone Mackin, spokeswoman for First Union, who until a few years ago worked at Bank of America.</P> <P><P> McColl was among the first to call Crutchfield when he was diagnosed with cancer, she said. And the two have been known to spent the evening talking after bumping into one another at parties or the country club.</P> <!–plsfield:end–> <P><CENTER> © 2001 The Washington Post Company </CENTER></P> <P><CENTER> <A HREF="/wp-dyn/business/A32068-2001Jan22.html"><img src="http://a188.g.akamaitech.net/f/188/920/1d/www.washingtonpost.com/wp-srv/images/arrow_previous.gif" width="14" height="12" border="0"> □□□8700□□□ <A HREF="http://www.washingtonpost.com/wp-srv/maps/mit_foto.map"><IMG SRC="http://a188.g.akamaitech.net/f/188/920/1d/www.washingtonpost.com/wp-srv/images/channelnav_news.gif" WIDTH="760" HEIGHT="16" BORDER="0" ALT="channel navigation" ISMAP="true"></A>
</TD> </TR><TR> <TD></TD> <TD><IMG SRC="http://a188.g.akamaitech.net/f/188/920/1d/www.washingtonpost.com/wp-srv/globalnav/images/spacer.gif" WIDTH="428" HEIGHT="1" BORDER="0" ALT=" "></TD> <TD></TD> <TD></TD> </TR></FORM></TABLE> <FONT SIZE="-2">
</FONT> </TD></TR></TABLE> </BODY></HTML>

Fig. 4. An HTML file of WPOST where frequent n-grams on $(24, 10)$ are colored with gray

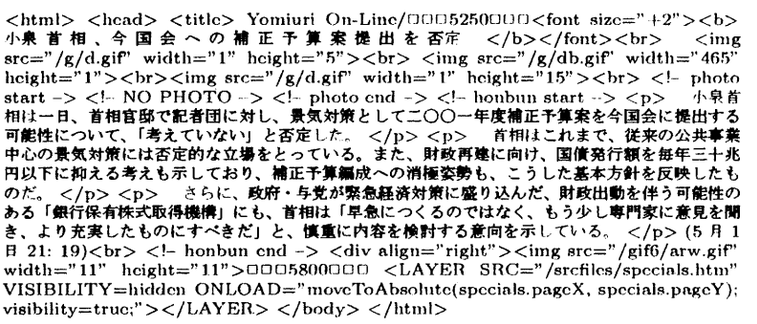


Fig. 5. An HTML file of YOMIURI where frequent n-grams on $(27, 10)$ are colored with gray

is about 3.2M Bytes (average 42K Bytes), the minimum size of the articles is 31K Bytes, and the maximum size is 65K Bytes.

Given WPOST, `FindOptimal` outputs $(24, 10)$ as an optimal cut point. Fig. 4 is an HTML file in WPOST where frequent n-grams on $(24, 10)$ are colored with gray. Some letters are omitted because the file is too large. A number surrounded by three boxes □, such as □□□5250□□□, denotes the approximative number of omitted letters. The color of omitted letters is the same as the color of the boxes and the number. In Fig. 4, the longest black substring completely equals to the body of the article except for the last period. Short black strings are substrings of the headline, which appear twice, or a substring of the file name in which this article is contained. A file name is unique in this set, so that a part of the file name is remains not to be colored.

The accuracy, recall, and precision on WPOST is 0.975, 0.939, and 0.872, respectively. The precision is relatively lower because unique substrings are included in the header of an HTML file such as a part of the file name.

The other set consists of articles written in Japanese. They are collected from "Yomiuri On-Line (`http://www.yomiuri.co.jp/`)". This is constructed by collecting all linked pages from all categories. For example, economic articles are collected from `http://www.yomiuri.co.jp/02/index.htm`. This set is denoted by *YOMIURI*. The number of the articles in YOMIURI is 198, the total size of YOMIURI is about 2.7M Bytes (average 13.4K Bytes), the minimum size of the articles is 297 Bytes, and the maximum size is 39K Bytes.

YOMIURI includes non-article files as noise data. The shortest file is not an article file. Moreover, files whose size are smaller than 12K Bytes are top pages of categories. Such files have only hyperlinks to articles. The number of noises in YOMIURI is 14 and its size is about 36K Bytes.

Given YOMIURI, `FindOptimal` outputs $(8, 10)$ as an optimal cut point. The accuracy, recall, and precision on YOMIURI is 0.992, 0.808, and 0.991, respectively.

Fig. 5 is a modified HTML file. Black parts are parts of the headline and the body of the article. The headline is at the second line and the content starts after "`<!-- honbun start -->`" which means the start of the body. Short colored substrings in the body are "<p>" tags with the beginnings or ends of sentences. These substrings decrease the recall. Ends of sentences appear frequently in a Japanese sentence and beginnings are frequent expressions in articles such as "prime minister (Koizumi)".

4.3 Articles from Different Sites

We use "Los Angeles Times (`http://www.latimes.com/`)" and "The Sankei Shimbun (`http://www.sankei.co.jp/`)" in addition to sites described in Section 4.2. Two sets of articles collected from the URLs are denoted by *LATIMES* and *SANKEI*, respectively. Articles in LATIMES and SANKEI are written in English and Japanese, respectively.

In this section, the following data sets are considered. Any combination of English and Japanese sites is included.

WP-LA articles from WPOST (30 articles 2.36M Bytes) and LATIMES (40 articles 2.44M Bytes)
SA-YO articles from SANKEI (23 articles 86K Bytes) and YOMIURI (198 articles 2.7M Bytes)
SA-LA articles from SANKEI (23 articles 86K Bytes) and LATIMES (150 articles 4.5M Bytes)

Note that the size of data set SANKEI is too small.

Table 1 shows outputs of `FindOptimal` and evaluation values. A cell $x(y, z)$ of an evaluation value shows the total value x on all articles and two evaluation values y and z on each site articles. The total value is not the average of the corresponding two single sites. It is calculated by just counting letters of all articles from both sites. That is, if two evaluation values of single site are x_1/y_1 and x_2/y_2, then the total evaluation is $(x_1 + x_2)/(y_1 + y_2)$. Therefore, in data sets SA-YO and SA-LA, the evaluation values are dominated by those of the large data set, YOMIURI or LATIMES, respectively.

Table 1. Evaluation values for different sites data

Data Set	Optimal	Accuracy	Recall	Precision
WP-LA	$(28, 17)$	0.965 (0.969, 0.962)	0.802 (0.906, 0.679)	0.894 (0.859, 0.954)
SA-YO	$(19, 13)$	0.994 (0.892, 0.997)	0.889 (0.697, 0.930)	0.987 (0.965, 0.990)
SA-LA	$(23, 6)$	0.959 (0.914, 0.961)	0.958 (0.992, 0.954)	0.755 (0.796, 0.749)

Articles of the same site have the same formats. Data sets in Table. 1 have different formats since documents are collected from different sites. `FindOptimal` detects different formats as useless parts with high accuracy. Especially, it detects useless parts of articles in SANKEI despite of its small size.

4.4 Discussion

Many text mining algorithms assume that frequent patterns, substrings, rules, etc. are important. But, in this paper, frequent n-gram is defined to be useless. `FindOptimal` avoid treating an important keyword as a useless substring in the following way. `FindOptimal` increases the length n and frequency a from the initial cut point $(2, 1)$. Therefore, a frequent n-gram turns out to be long. In fact, $n = 24$ when WPOST is the input. We think that an important keyword is not so long and a substring for the structure and style is relatively long. Even if some important keywords are long, it merely happens that they appear frequently. Thus, the algorithm does not judge an important keyword to be useless.

This simple idea sometimes does not work well if the size of given input documents are not enough. For example, in YOMIURI data set, there exist some news files of Mr. Tsuta's death. He was a famous high-school baseball manager. In these news files, the substring "a former manager of the baseball club at Tokushima prefectural high-school" appears frequently. In Japanese, the length of the substing is 11, and the length of the cut point found by `FindOptimal` is 8. So, `FindOptimal` treats the substring to be useless although the substring is not for the structure or style of the documents. This type of errors does not happen when enough articles are given to `FindOptimal` because the frequency of such long substring become to be relatively low.

Note that useless parts do not consist of only tag sequences. For example, articles in YOMIURI have the same string in <title> tag and the string is detected as a useless part. On the other hand, the title of an article in WPOST is the same as the headline of the article and the title is detected as a non-useless part.

5 Conclusion

We introduced a new static value *alternation count* and developed an algorithm that divides each document of given documents into two parts, useless parts and non-useless parts. It is based on a simple assumption that frequent n-grams are not useful. We defined an optimal cut point to decide an appropriate pair (n, a) of length n and frequency a.

The algorithm does not depend on natural and markup languages because it counts substrings of inputs instead of words or other grammatical units. Experimental results show that the algorithm is robust for noise. Moreover, if input documents are collected from different sites and have different formats, an outputed cut point divides documents of both sites into useless and non-useless parts with hight accuracy.

We only showed experiments on news articles provided as HTML files. It is a future work to use other types of data sets such as dynamic pages, static pages except for news, files written in other markup language, etc.

It is an interesting future work to use some knowledge on grammars of the markup language. For example, when the algorithm enumerates n-grams, if a

delimiter such as "<", ">", ".", and "?", is contained in an n-gram, they must be at the beginning or end of the n-gram. This may improve accuracies.

As an application of the algorithm, we developed a record extraction system SCOOP [16]. A record extraction is an important application of Web mining [6, 9,12]. SCOOP utilized `FindOptimal` as the preprocessor and knowledge that a delimiter of a record (or field) ends with ">" and begins with "<". Ginve an output of `FindOptimal`, SCOOP searches delimiters only near boundaries of non-useless parts and outputs the most frequent pair of substrings as a delimiter if it is unique on each record.

Another challenging future work is to apply our algorithm to genome infomatics. The *longest common subsequence problem* is, given two string, to find a longest common subsequence of them [2,8]. The problem for k ($k \geq 2$) strings is known as *multiple sequence alignment*, which is a major problem in genome infomatics [11]. If k is not fixed, multiple sequence alignment is known to be NP-complete [13]. Both multiple sequence alignment and our problem are to extract parts common to a given set of strings. A difference between two problems is that each common part should have particular length in our setting, that is, our algorithm does not work well if common parts are too short.

References

1. S. Abiteboul, Querying Semi-structured Data. In *Proc. of the 6th International Conference on Database Theory*, pp. 1–18, 1997.
2. A. V. Aho, D. S. Hirschberg and J. D. Ullman, Bounds on the Complexity of the Longest Common Subsequences Problem. *J. ACM*, Vol. 23, No. 1, pp. 1–12, 1976.
3. R. Agrawal, T. Imielinski and A. Swami, Mining Association Rules between Sets of Items in Large Databases. In *Proc. of the 1993 ACM SIGMOD Conference on Management of Data*, pp. 207–216, 1993.
4. R. Agrawal and R. Srikant, Mining Sequential Patterns. In *Proc. of the 11th International Conference on Data Engineering*, pp. 3–14, 1995.
5. H. Arimura and S. Shimozono, Maximizing Agreement with a Classification by Bounded or Unbounded Number of Associated Words. In *Proc. of the 9th International Symposium on Algorithms and Computation*, Lecture Notes in Computer Science 1533, pp. 39–48, 1998.
6. P. Atzeni and G. Mecca, Cut and Paste. In *Proc. of the 16th ACM SIGMOD Symposium on Principles of Database Systems*, pp. 144–153, 1997.
7. CompletePlanet, The "Deep Web" White Paper. `http://www.completeplanet.com/Tutorials/DeepWeb/index.asp`.
8. M. Crochemore and W. Rytter, *Text Algorithms*. Oxford University Press, New York 1994.
9. D. W. Embley, Y. Jiang and Y. -K. Ng, Record-Boundary Discovery in Web Documents. In *Proc. of the 1999 ACM SIGMOD Conference*, pp. 467–478, 1999.
10. W. B. Frakes and R. Baeza-Yates (eds.), *Information Retrieval: Data Structures & Algorithms*. Prentice-Hall, 1992.
11. D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York, 1997.

12. N. Kushmerick, D. S. Weld and R. B. Doorenbos, Wrapper Induction for Information Extraction. In *Proc. of the 15th International Joint Conference on Artificial Intelligence*, pp. 729–737, 1997.
13. D. Maier, The Complexity of Some Problems on Subsequences and Supersequences. *J. ACM*, Vol. 25, No. 2, pp. 322–336, Apr. 1978.
14. H. Mannila and H. Toivonne, Discovering Generalized Episodes Using Minimal Occurrences. In *Porc. of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 146–151, 1996.
15. M. F. Porter, An Algorithm for Suffix Stripping. *Automated Library and Information Systems*, Vol. 14, No. 3, pp. 130–137, 1980.
16. Y. Yamada, D. Ikeda and S. Hirokawa, SCOOP: A Record Extractor without Knowledge on Input. In *Proc. of the 4th International Conference on Discovery Science*, Lecture Notes in Artificial Intelligence, 2001. (to appear)

Multicriterially Best Explanations

Naresh S. Iyer and John R. Josephson

The Ohio State University,
Laboratory for Artificial Intelligence Research,
Computer and Information Science Department,
Columbus, Ohio, 43210 USA
{niyer,jj}@cis.ohio-state.edu

Abstract. Inference to the best explanation, IBE, (or abduction) requires finding the best explanatory hypothesis, from a set of rival hypotheses, to explain a collection of data. The notion of *best*, however, is multicriterial and the available rival hypotheses might be variously good according to different criteria. Thus, one can view the abduction problem as that of choosing the best hypothesis from among a set of multicriterially evaluated hypotheses - i.e as a *multiple criteria decision making* problem. In the absence of a single hypothesis that is the best along all dimensions of *goodness*, the MCDM problem becomes especially hard. The Seeker-Filter-Viewer architecture provides an effective and natural way to use computer power to assist humans to solve certain classes of MCDM problems. In this paper, we apply an MCDM perspective to the abductive problem of red-cell antibody identification and present the results obtained by using the S-F-V architecture.

1 Introduction

Abductive inference is a ubiquitous form of reasoning in science and common sense. Abduction has been referred to as *inference to the best explanation* by Harman [3] and as *the explanatory inference* by Lycan [4]. Typically the available evidence is insufficient to narrow conclusively to single explanations. So, multiple hypotheses are available and the problem becomes one of choosing the best among rivals. Josephson & Josephson [2] have described abductions as following this pattern:

D is a collection of data (facts, observations, givens)
H explains D (would, if true, explain D)
No other hypothesis can explain D as well as H does.

Therefore, H is probably true.

They also suggest that the judgment of likelihood associated with a conclusion should depend upon a number of considerations. Apart from how good a single hypothesis is by itself, it is also desirable that it decisively surpass the

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 128–140, 2001.

alternative hypotheses. However, there are in general, multiple kinds of criteria by which hypotheses may be compared. Explanatory power and plausibility are examples. Thus, we may view abduction as requiring a choice among the multicriterially evaluated hypotheses, that is a species of multiple criteria decision making.

MCDM problems have been widely studied across diverse fields and many techniques abound for solving MCDM problems [5]. An important concept in MCDM is the idea of dominance. Dominance is very much like an *all-other-things-being-equal* kind of reasoning. Specifically, we say that some multicriterially evaluated alternative A *dominates* another alternative B if there is some criterion in which A is strictly better than B and there is no criterion in which B is strictly better than A. An alternative that is not dominated is called a Pareto Optimal alternative. For a given problem, the set of Pareto Optimal alternatives has the property that, within the set, the only way to improve along any dimension is to accept a loss in another dimension. That is, choosing among the Pareto Optimal alternatives is a matter of making trade-offs. It is known that the size of the Pareto-optimal set is typically a very small percentage of the actual number of alternatives [6] [7]. Thus, the application of dominance as a *filter* can be expected to considerably reduce the number of alternatives which need to be considered [1].

It is worth noting is that there is no loss incurred in the elimination of the dominated alternatives unless significant criteria have not been considered. This is because we know that for every alternative eliminated by the dominance filter, there is at least one Pareto-optimal alternative that dominates it and is therefore *multicriterially better* than it. The application of dominance minimally requires that an order relation hold among values for each criterion. The survivors of the dominance filter represent the multicriterially maximal subset of alternatives from the original set.

From the definition of dominance it is clear that, for each pair of alternatives that survive the dominance filter, they outperform each other according to different criteria. In other words, if alternatives A and B are in the Pareto Optimal set, then it must be the case that there is at least one criterion in which A is better than B and that there is at least one criterion in which B is better than A, thereby preventing either from dominating the other.

In an abduction problem, a more plausible hypothesis, H_1, might not explain as much as a less plausible one, H_2. That is, H_2 is better according to explanatory coverage while H_1 is better according to the criterion of plausibility. In such a case, there is no obvious sense in which either H_1 or H_2 can be said to be a distinctly better hypothesis. However, depending upon the need to explain more, and upon the degree of confidence that is needed for the final choice, a choice between H_1 or H_2 may become possible. The choice from among the Pareto Optimal set requires that trade-offs be accepted between plausibility and explanatory coverage. This can be a challenge since such trade-off judgments are often a function of the specific values at hand. For example, a certain level of confidence or of explanatory power may be sufficient.

In summary, a general way to solve an MCDM problem is to apply the dominance filter and then allow for choice from among the set of dominance survivors by applying human trade-off judgments with respect to the various criteria. The *Seeker-Filter-Viewer* architecture described in [1] is based on this strategy for solving the MCDM problem. The *Seeker* is a module which generates applicable alternatives and produces evaluations for them according to the different criteria. The *Filter* uses the principle of dominance to produce the Pareto-optimal set from the generated and evaluated set of alternatives. It eliminates the distinctly suboptimal alternatives. The *Viewer* allows a human to express his trade-off judgments on the Pareto-optimal alternatives. The *Viewer* allows a user to view the candidate alternatives as points in graphs with the criteria as axes. If multiple criteria need to be considered, the Viewer will provide multiple interlinked 2-D plots and histograms. The human expresses preferences by selecting desirable regions in the graphs. The graphically selected points or regions are cross-linked across all the open plots so that a selection made on one plot shows the values of the selected alternatives according to the other criteria.

Apart from explanatory coverage and plausibility, we will describe several other criteria that can generally be used to evaluate candidate hypotheses in abduction problems. These criteria may or may not apply depending upon the problem domain and other characteristics of the data. We will briefly describe the S-F-V architecture and as an illustration both of viewing abduction from an MCDM perspective, and a demonstration of applying the S-F-V architecture, we will present the results of experiments in the domain of red cell antibody identification as described in [2]. We will describe the antibody identification problem as an abduction problem, and define the evaluation criteria used in the experiment. Finally, we will show the results of viewing this abduction problem as an MCDM problem and applying the S-F-V architecture to help solve the problem.

2 The Seeker-Filter-Viewer Architecture

The S-F-V architecture is described in detail in [1] and [9]. In this section, we provide a brief overview of the architecture and its use in solving MCDM problems. Essentially, the architecture is composed of three modules, the *Seeker*, the*Filter*, and the *Viewer*, each designed to perform a specific set of functions involved in solving the given MCDM problem. We next describe these components one at a time:

2.1 The Seeker

The Seeker is responsible for the generation of the choice alternatives for the MCDM problem. In case the choice alternatives are already present or supplied by the decision-maker, the Seeker makes these choice alternatives accessible to the Filter by reading them from the database. For problems where the decision-maker cannot provide the choice alternatives himself, it is the function of the

Seeker to seek out the alternatives from whatever sources are available, in a form that can be used by the Filter. Abstractly, this could be a search on the Internet looking for choice alternatives pertaining to the problem. The Seeker described in [1] is currently capable of generating choice alternatives as compositions of various components listed in a component library. The Seeker instantiates all possible choice alternatives that can be formed by some distinct composition of a set of components in the library. Having instantiated the choice alternative, it next makes use of simulation models to evaluate various property values for the choice alternatives. For example, for an instantiated *car*, the Seeker might run simulations to compute the *mileage, cost, weight, top-speed* and other properties related to cars, for which simulation models are available. At the end of the generation process, the Seeker produces a list of choice alternatives along with a set of {*property-name, property-value*} pairs for each alternative. It makes this list available to the Filter.

2.2 The Filter

The Filter is responsible for applying the dominance rule to the set of alternatives generated by the Seeker. In order to do this, the Filter expects the decision-maker to choose those properties of the choice alternatives which reflect the dimensions of outcomes that matter to him, and additionally the directions of goodness for the criteria. For example, if the decision-maker desires to buy a car that is cost-effective to him, he should choose *cost* and *mileage* as properties of interest to him. Once such a set of properties have been selected by the decision-maker, the Filter uses these properties as criteria based on which to apply the dominance rule on the set of alternatives. Since the criteria values for the chosen criteria are already made available by the Seeker, the Filter makes use of these values to produce the Pareto-optimal set of alternatives. As mentioned earlier, this step is essential because alternatives not belonging to the Pareto-optimal set are known to be dominated by some Pareto-optimal alternative. As a result, there is no loss incurred in eliminating such alternatives. By doing so, the Filter prevents the decision-maker from having to even consider such alternatives, and thereby unintentionally select a suboptimal alternative. Finally, as indicated in [6], [7], the Pareto-optimal set often tends to be a very small fraction of the original set. Hence the application of the Filter also reduces the size of the set of alternatives that further need to be considered. While the Filter can reduce the relevant set of alternatives from a large number to a small fraction, choosing an alternative even from a handful of Pareto-optimal alternatives can be a demanding task for the decision-maker. The next module of the architecture allows the decision-maker to graphically interact with the Pareto-optimal alternatives in various ways, in order to select the final choice alternative(s) of interest to him.

2.3 The Viewer

As mentioned previously, choice among Pareto-optimal alternatives requires the making of tradeoffs. The Viewer allows the decision-maker to interact with the

Pareto-optimal set by means of various kinds of graphical plots which enable the decision-maker to express his tradeoff preferences in the context of the available alternatives. A more detailed description of all modes of interaction that the Viewer allows, along with a description of an interaction session between the Viewer and a decision-maker is provided in [9]. Here will only mention that the Viewer allows the decision-maker to plot the Pareto-optimal alternatives as points in 2-D scatter plots where the axes of the plots can be selected by the decision-maker himself; he can further pull up as many plots as he desires. The Viewer also maintains a set of 1-D plots where the Pareto-optimal alternatives are plotted along single property-axes. The Viewer allows the decision-maker to select points or collections of points by enabling graphical selection of such points. Upon selection by the decision-maker, all points within the selected region are indicated using a separate color and moreover such indication is provided across all the open points. Thus ,even though the decision-maker makes his selection on a single plot, he gets to examine the implications of his selection in terms of the other properties by examining the colored points on all other plots. This forces the decision-maker to make selections and at the same time evaluate the consequences of the selection. It is expected that this will lead to a more rational selection process. Apart from making tradeoffs, the Viewer enables other kinds of preference expression by the decision-maker. These include: choosing alternatives by categories from bar-charts, applying hard-constraints based on criteria by using the 1-D plots, applying various kinds of constraints based on as yet unconsidered properties, combining alternatives that belong to different Viewer-based selections of the decision-maker, looking at a list of all properties of alternatives in the selected region in a tabular form, and so on. Thus, the Viewer complements the Filter by enabling many kinds of preferences that apply when choosing from Pareto-optimal alternatives.

The synergy between the three modules of the S-F-V architecture provides it with the ability to act as an effective decision support for solving MCDM problems. As an indication of its effectiveness, we point to the experiment described in [1] where close to 2 million choice alternatives (Hybrid vehicles) were generated by the Seeker, the Filter reduced this set to 1078 alternatives, and interaction between a decision-maker and these Filter survivors using the Viewer resulted in a final output of 7 alternatives. The architecture has been applied to a number of engineering problems and our claims about the effectiveness of the architecture are based on the response we received from the users regarding the ease with which they were able to use the architecture. We realize that a formal usability analysis of the architecture would go a long way towards establishing this. We have compared the Viewer with a few other alternative visualization techniques in MCDM literature and our impression is that the Viewer has its own set of unique properties. We direct the interested reader to a survey of such visualization techniques that occurs in [10] (pp. 238-249).

This brings our description of the S-F-V architecture to a close. We next describe some properties of explanatory hypotheses, which can be used as evaluation criteria for the hypotheses. The use of such criteria to evaluate hypotheses

will allow hypotheses to be viewed as multicriterially evaluated alternatives, thereby allowing the problem of choosing the "multicriterially best" alternative to be seen as an MCDM problem.

3 Evaluation Criteria for Explanatory Hypotheses

As we said, the idea of the *best* hypothesis from among a set of hypotheses is a multicriterial notion. In [8] the following qualities are suggested as criteria for evaluating hypotheses: *Explanatory Power*, *Plausibility*, *Internal consistency*, *Simplicity*, *Specificity*, *Predictive Power*, and *Theoretical Promise*. In order to apply MCDM techniques it will be necessary that the evaluations according to the criteria can be obtained in a numerical form, or some other form that enables the comparison of criterion values, so that it is conducive to the application of MCDM techniques. This may well depend upon the domain for which the abduction problem is being solved. As an illustration of how this can be done, we next describe how evaluations were produced for the hypotheses in red cell antibody identification domain.

4 The RED Domain: The Red Cell Antibody Identification Task

As described in [2], the RED systems are medical test-interpretation systems that operate in the knowledge domain of hospital blood banks. Specifically, the RED systems are meant to help in the problem of red-cell antibody identification. We will first briefly describe the problem and then formulate the problem as an abduction problem.

4.1 The Problem

Before blood transfusion is carried out it is imperative to check that the donor's blood matches the patient's blood. The process of matching involves ensuring that the donor's blood does not contain antigens which would be identified as foreign bodies by the patient's immune system. If the immune system does encounter foreign bodies, it produces antibodies directed against them. The antibodies that are produced by the patient's blood against red cell antigens of a donor are called red-cell antibodies. If the patient's blood contains antibodies directed against the red cell antigens of the donor's blood, this is a case of mismatch. Transfusion of badly matched blood could result in many bad consequences including fever, anemia, and life threatening kidney-failure. Hence the red cell antibody identification task is of crucial importance to blood banks. In addition to the familiar A, B, and Rh, more than 400 red-cell antigens are known. Once the blood has been tested to determine the patient's A-B-O and Rh blood type, it is necessary to test for the presence of antibodies directed toward other red-cell antigens.

Table 1. Red-cell test panel. The various test conditions, or phases, are listed along the left side (AlbuminIS, etc.) and identifiers for donors of the red cells are given across the top (623A, etc.). Entries in the table record reactions graded from 0, for no reaction, to 4+ for the strongest agglutination reaction, or H for hemolysis. Intermediate grades of agglutination are +/- (a trace of reaction), 1+w(a grade of 1+, but with the modifier "weak"), 1+, 1+s(the modifier means "strong"), 2+w, 2+, 2+s, 3+w, 3+, 3+s, 4+w. Thus, cell 623A has a 3+ agglutination reaction in the Coombs phase.

	623A	479	537A	506A	303A	209A	186A	195	164
AlbuminIS	0	0	0	0	0	0	0	0	0
Albumin37	0	0	0	0	0	0	0	2	1
Coombs	3+	0	3+	0	3+	3+	3+	3+	3+
EnzymeIS	0	0	0	0	0	0	0	0	0
Enzyme37	0	0	1+	0	0	1+	0	1+	0

Typically this identification is performed by using one or more reaction panels of the form shown in Table 1. The columns in the table refer to different applicable donors, while the rows refer to different test conditions. Each entry in the table indicates reactions shown by a mixed sample of the patient's blood serum and the indicated donor's red blood cells, under the specified test conditions. These figures are produced by the blood bank technologist to indicate his visual assessment of the strength and type of reaction. Possible reaction types are agglutination (clumping of cells) or hemolysis (splitting of the cell walls). The strength of the reactions are expressed in the blood-banker's vocabulary, some terms of which are shown in Table 1, and consists of thirteen possible reaction strengths. Hemolysis reactions were ignored for purposes of this experiment. All 3+ entries are converted into the number 3 for our experiment. Similarly, the 1+ values are converted to number 1 and so on. Reactions indicated as $2 + s$ are converted into the number 2.5 while those marked as $2 + w$ are converted into the number 1.5.

Additionally, information about the significant antigens present in each of the donor samples are recorded in a table called the antigram. By reasoning about the pattern of reactions displayed by the reaction panel and using the antigen information present in the donor antigram, the blood-bank technologist attempts to determine which antibodies are present in the patient's serum and are causing the observed reactions and which are absent, or at least not present in enough strength to cause reaction. The RED systems were built to automate this reasoning process.

4.2 The Red Cell Antibody Identification Problem as an Abduction Problem

The reaction panel shown in Table 1 can be considered as data to be explained. Using the antigrams which give information about the various antigens present in the donor samples, it is possible to construct hypotheses about the existence

of various antibodies in the patient's serum. Each such hypothesis will contain two kinds of information -

- A profile similar to Table 1 representing how much this particular hypothesis can offer to explain for each of the reactions in the panel. This is the most that can be consistently explained by the hypothesis.
- A plausibility value, which is the result of applying rules given by domain experts, to the data of the case. In our experiment, this value is an integer between -3 to +3, representing the plausibility on a symbolic scale from "ruled out" to "highly plausible".

An example for a certain antibody is given in Table 2. It shows how much of the reactions shown for the case from Table 1 are accounted for by hypothesizing that the antibody, AntiNMixed, is present in the patient's serum. Also, the plausibility value for the hypothesis is indicated to be -2.

Table 2. Reaction profile for an individual antibody(Anti NMixed) hypothesis. Note by comparison with the overall reaction panel in Table 1 that the hypothesis only offers to partially explain some of the reactions.

Anti NMixed Profile; Plausibility=-2						
	623A	479	303A	209A	186A	195
AlbuminIS	0	0	0	0	0	0
Albumin37	0	0	0	0	0	0
Coombs	0.5	0	2	0.5	0.5	2
EnzymeIS	0	0	0	0	0	0
Enzyme37	0	0	0	0.5	0	0.5

Table 2 does not contain as many columns as Table 1 because the hypothesis cannot explain any of the reactions pertaining to those columns. The same kind of profile is created for all of the other non-ruled out antibodies. Hence given a donor, the following inputs are present:

1. The reaction panel as indicated in Table 1
2. A plausibility value for each antibody.
3. A reaction profile for each antibody for which the plausibility value is not -3, i.e. it has not been "ruled out."

The desired output will be a set of antibodies which best explain the reactions, along with plausibility values associated with them. The above problem can now be seen as an abduction problem with the following mapping:

1. The reaction panel represents the data, D, to be explained.
2. The individual antibodies which have not been "ruled out" and all possible composite hypotheses that can be generated from them represent the set of possible explanatory hypotheses, the set E.

The abduction problem is one of finding the hypothesis which *best* explain the reactions in the reaction panel. However, sometimes the evidence will be insufficient and there will be no unique, best explanation.

5 Evaluation Criteria for Hypotheses in the RED Domain

In this section, we will describe how the evaluation criteria for the hypotheses in the RED domain were computed from the given information. For a given problem, the set E of all possible explanatory hypotheses was created as described next.

Firstly, the set of antibodies which are ruled out(i.e. with plausibility values -3) are no longer considered as potentially explanatory hypotheses for the problem. Such hypotheses are excluded from set E. The set, S, of simple hypotheses may be defined as follows:

$$S = \{\ A_i : A_i \textit{ hypothesizes the presence of a particular antibody and the plausibility of } A_i \textit{ is not } -3\}$$

Now, the set E of applicable hypotheses is defined as the set of all possible hypotheses obtainable as combinations(conjunctions) of the simple hypotheses in S.

The set $C = E - S$ is therefore the set of all *composite hypotheses* which hypothesize the presence of more than a single antibody to explain the reactions. Thus, if we suppose $A_1, A_2, A_3, \cdots, A_k$ to be each of the individual hypotheses related to the presence of single antibodies which have not been ruled out in advance, then the hypothesis A_4 is an example of a *simple hypothesis* while the hypothesis $\{A_2, A_3, A_k\}$ is an example of a 3-part *composite hypothesis*. Obviously, the size of the largest composite hypothesis is k and it includes *all* of the simple hypotheses in the set S as its parts. Next, we will discuss how some of the criteria mentioned in Section 3 were computed for the set E above.

1. *Explanatory Power*: Given the values in the reaction panel and the reaction profiles, R_i, for simple hypotheses, A_i, one way to quantify the explanatory power of a simple hypothesis is to compute the sum of all the values in its reaction profile table. Since each individual entry in the reaction profile offers to explain an observed reaction as consistently as possible, the sum of the reaction profile matrix is indicative of the overall explanatory power of the simple hypothesis. This value is used as a heuristic measure of the explanatory power of the simple hypothesis. For a composite hypothesis, the reaction profile is constructed by using the profiles for its parts. That is, the reaction profile for a composite hypothesis is constructed as the entry-wise sum of reactions in the individual reaction profiles, with a maximum for any entry of the reaction strength that needs to be explained. More formally the Explanatory Power, $\mathcal{E}$, was computed as,

$$\forall H \in E, \mathcal{E}(H) = \sum_{a,b} R(a,b) \tag{1}$$

2. *Implausibility*: The plausibility, p_i, for each of the simple hypotheses, A_i, is already available as a part of the input. Since -3 is the lowest degree of plausibility that is assigned to a simple hypothesis, the implausibility can be computed by a heuristic measure which produces low value of implausibilities for high value of plausibilities and so on. The exact form of this function is not important since the values themselves are meant to be used only for relative comparisons. In our experiment, implausibility, $\mathcal{I}$, was computed as,

$$\forall H \in E, \mathcal{I}(H) = \sum_{A_j \in H} (4 - p_j)$$

3. *Simplicity*: There are at least two ways to define this criterion.
   a) *Cardinality*: Simplicity can be defined in terms of the number of parts in a hypothesis, that is its cardinality. Note that we would want to minimize this value in order to maximize simplicity. However, this measure provides a better score for a hypothesis like $\{A_2, A_6\}$ relative to another hypothesis like $\{A_1, A_3, A_7\}$ based merely on the differences in their structural simplicities.
   b) *Inclusion simplicity*: This measure cannot be quantified on a per hypothesis basis like the previous ones. However, when comparing two composite hypotheses, say H_1 and H_2, we say that H_1 is better in inclusion-simplicity than H_2 if and only if all of the constituent parts of H_1 are present in H_2 as well. In all other cases, the two hypotheses are considered incomparable in simplicity. This measure makes sure that the least complex hypothesis is preferred to a more complex one that explains no more.

For the RED domain experiment, only two of the above criteria were used. This is because the implausibility value, as defined previously, already carries the information carried by the inclusion-simplicity criterion. The addition of another hypothesis to a composite will always reduce its plausibility. So an *included* hypothesis will always be more plausible than the *including* one. Consequently, if k simple hypotheses are not ruled out in advance, then the abduction problem involves as many hypotheses as the total possible combinations that result from k, simple hypothesis. In other words, the problem becomes a $(2^k - 1)$ alternative, 2-criteria, MCDM problem. In the next section, we discuss the results of applying the S-F-V architecture to this MCDM problem.

6 Results of Applying the S-F-V Architecture

It is to be noted that the potential number of explanatory hypotheses in the RED domain is exponential in the number of simple hypotheses that have not been ruled out at the onset. Considering that up to 30 clinically significant red-cell antigens are known, the total number of alternatives is potentially quite large.

Hence, the ability of dominance filter to prune effectively becomes valuable in reducing the complexity of the problem. The results shown below are for the case labeled OSU-9 in the RED domain as described in [2]. The reaction panel shown in Table 1 refers to the same case.

This case resulted in 15 simple hypothesis which could not be ruled out based on the evidence at the outset. Thus we have a total of 32,767 potential explanatory hypotheses, a formidable number. The Seeker generates this set by building the exhaustive set of combinations starting with 15 simple hypotheses. In the process of generation, the Seeker also evaluates the hypotheses along the various criteria using the heuristic measures indicated in the previous section. It now makes this set of multicriterially evaluated hypotheses available to the Filter. The Filter, after applying the dominance rule using implausibility and explanatory power as the criteria produces a Pareto-Optimal set containing only 3 hypotheses! In other words, as long as the goal is to find the most plausible hypothesis which explains the reactions the best, there is no need to consider the remaining 32764 eliminated alternatives; dominance ensures that they are inferior to the survivors. The remaining 3 surviving hypotheses are plotted as points in a *Viewer* scatter plot shown below with the Implausibility and Explanatory Power as the axes. The labels for each point show the composition

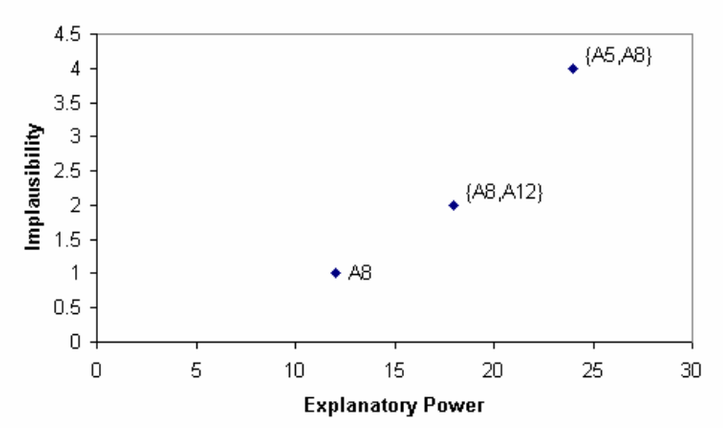


Fig. 1. Plot showing the 3 survivors of dominance applied to the case OSU-9 from the RED domain

of the individual hypotheses. We see that of the 3 survivors, one is a simple hypothesis and in fact this hypothesis, $A8$ occurs in each of the remaining two composite hypotheses, $\{A8, A5\}$ and $\{A8, A12\}$.

Figure 1 also shows the trade-offs available to the user of such a system. Such a trade-off is typical of many abduction problems where the ability to explain more comes with a cost in the confidence associated with the explanation. By

using this plot, which is displayed by the *Viewer* in the S-F-V architecture, the user can exercise his trade-off judgments by selecting the point of interest to him. For example, to get greater explanatory coverage than that provided by the simple hypothesis, the user is informed from the plot that he will need to incur an increase in the implausibility. The composite $\{A8, A12\}$, shown as the middle point in the plot, allows for one step of trade-off in the direction of explaining more, with a resulting increase in the implausibility. Similarly, the point to the extreme right and top explains the most but is also the most implausible among the three potential explanatory hypotheses. Figure 2 shows similar trade-offs for another experimental case. This plot shows more clearly, how moving from the leftmost point to the next point results in a considerable increase in explanatory coverage while the resultant increase in implausibility is not as large. Conversely, looking at the rightmost pair of points, we see that a very small increase in explanatory coverage is obtained by incurring quite large increase in implausibility. This illustrates how the different kinds of trade-off judgments can be brought upon to choose between competing hypotheses even if they are both Pareto Optimal. This plot also shows how choice between

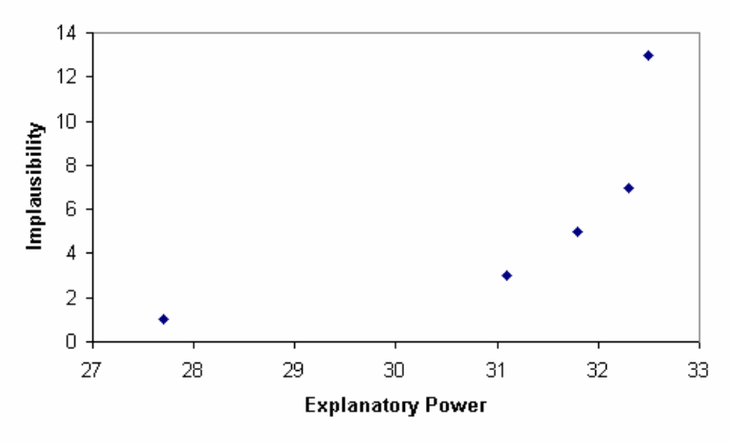


Fig. 2. Plot showing the survivors of dominance applied to the case Pat-32 from the RED domain

multicriterially best explanations involves trade-off. The choice of an appropriate hypothesis will depend upon the user's (in this case the person administering the blood) willingness to hypothesize the presence or absence of an antibody according to the urgency of the situation and other risk based considerations. Alternatively, if additional knowledge becomes available at a later stage of the problem, this may be used to rule out some of the surviving hypotheses.

7 Conclusions

We have shown how the MCDM perspective applies to abductive reasoning. IBE problems are inherently multicriterial. These criteria need not be commensurable. Even if that is the case, a well-defined notion of *multicriterially best explanations* can be given. Such best explanations need not be unique. However computer-aided visualization of the alternatives can help human to choose from among the multicriterially best hypotheses. It is worth noting that if there is indeed a single hypothesis that is the most plausible, explains the most, and so on, then such a hypothesis will be the sole survivor of the dominance filter (this is because by virtue of being the *best* along *all* of the evaluation criteria, it will dominate every other alternative, using the definition of dominance from page 2). Moreover MCDM techniques can help reduce the complexity of the problem. One can envision scientists using powerful, computerized decision aids like the S-F-V architecture in the future to help solve complex problems of discovery.

Acknowledgments. This material is based upon work supported by The Office of Naval Research under Grant No. N00014-96-1-0701. The support of ONR and the DARPA RaDEO program is gratefully acknowledged. Standard disclaimers apply.

References

1. Josephson, John, R., Chandrasekaran, B., Carroll Mark, Iyer Naresh, Wasacz Bryon, Rizzoni Giorgio, Li Qingyuan, Erb, David A. An Architecture for Exploring Large Design Spaces. Proceedings of National Conference of the American Association for Artificial Intelligence, Madison, Wisconsin, pp 143-150 (1998)
2. Josephson John, R., Josephson, Susan, G.: Abductive Inference: Computation, Philosophy, Technology. Cambridge University Press (1994)
3. Harman, G.: The Inference to the Best Explanation. Philosophical Review, Vol. 74, pp. 88-95, (1965)
4. Lycan, W., G.: Judgement and Justification. Cambridge University Press (1988)
5. Keeney, Ralph, L., Raiffa Howard: Decisions with multiple objectives: preferences and value tradeoffs. Wiley Publishers (1976)
6. Calpine H. C., Golding A.,: Some properties of Pareto Optimal Choices in Decision Problems. International Journal of Management Science, Vol. 4, No. 2, pp. 141-147 (1976)
7. Bentley, J., L., Kung, H., T., Schkolnick, M., Thompson, C., D.: On the Average Number of Maxima in a Set of Vectors and Applications. Journal for the Association of Computing Machinery, Vol. 25, No. 4, pp. 536-543, (1978)
8. Josephson, John, R.,: Abduction-Prediction Model of Scientific Discovery Reflected in a Prototype System for Model-Based Diagnosis. Philosophica, Vol. 61, No. 1, pp. 9-17 (1998)
9. Chandrasekaran B.,: Functional and Diagrammatic Representation for Device Libraries. Technical Report, The Ohio State University, (2000)
10. Mietinnen, K. M.,: Nonlinear Multiobjective Optimization. International Series in Operations Research and Management Science, Kluwer Academic Publishers, (1999)

Constructing Approximate Informative Basis of Association Rules

Kouta Kanda, Makoto Haraguchi, and Yoshiaki Okubo

Division of Electronics and Information Engineering
Hokkaido University
N-13 W-8, Sapporo 060-8628, JAPAN
{ makoto, yoshiaki}@db-ei.eng.hokudai.ac.jp

Abstract. In the study of discovering *association rules*, it is regarded as an important task to reduce the number of generated rules without loss of any information about the significant rules. From this point of view, Bastide, et al. have proposed to generate only *non-redundant* rules [2]. Although the number of generated rules can be reduced drastically by taking the redundancy into account, many rules are often still generated. In this paper, we try to propose a method for reducing the number of the generated rules by extending the original framework. For this purpose, we introduce a notion of *approximate generator* and consider an *approximate redundancy*. According to our new notion of redundancy, many non-redundant rules in the original sense are judged redundant and invisible to users. This achieves the reduction of generated rules. Furthermore, it is shown that any redundant rule can be easily reconstructed from our non-redundant rule with its *approximate* support and confidence. The maximum errors of these values can be evaluated by a user-defined parameter. We present an algorithm for constructing a set of non-redundant rules, called an *approximate informative basis*. The *completeness* and *weak-soundness* of the basis are theoretically shown. Any significant rule can be reconstructed from the basis and any rule reconstructed from the basis is (approximately) significant. Some experimental results show an effectiveness of our method as well.

1 Introduction

The discovery of *association rules* is an important task in the research area of *Data Mining*. Its main purpose is to identify relationships among items in a given large database. This kind of problem has firstly introduced by Agrawal, et al. [1]. According to their statement, the problem can be divided into two sub-problems:

Finding frequent itemsets:
Given a transaction database $\mathcal{D}$, we try to find all *frequent* itemsets [1] in $\mathcal{D}$.

Generating confident association rules:
All *confident* association rules are generated based on the frequent itemsets.

[1] An itemset is a set of items appearing in $\mathcal{D}$.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 141–154, 2001.

In order to solve the former problem, we would be required to search in an *itemset-lattice* consisting of 2^m itemsets if we have m possible items. On the other hand, the latter problem can be solved in a straightforward manner, once we have all frequent itemsets. Therefore, the former is considered *primary* and the latter, *secondary* in an efficient discovery of association rules. In fact, many studies on association rule discovery have tended to concentrate on an efficient computation of the frequent itemsets and many algorithms for this task have been proposed [1,4,5].

Thus, as many researchers have actually investigated, the task of finding all frequent itemsets is one of the important subjects in the discovery of association rules. However, we still have another significant issue to be addressed. It is concerned with the number of rules generated from the obtained frequent itemsets.

In general, a large number of rules are generated and then presented to a user. Although it is ensured that the generated rules meet the requirements for *support* and *confidence* given by the user [2], they often include many rules that are not so interesting to the user in fact. Therefore, the user has to check each presented rule carefully in order to obtain *actually* interesting ones. However, such a task is quite hard due to the large number of presented rules. In some cases, unfortunately, several interesting rules might be missed. Therefore, it is helpful for the user to reduce the number of generated (and presented) rules without loss of any information of possible ones. The purpose of this paper is to propose a method for such a reduction.

By introducing a notion of *redundancy* of association rules, Bastide, et al. have proposed to identify only the set of *non-redundant* ones, called an *informative basis*, and to present the basis to the user. In a word, a non-redundant rule can be viewed as a *representative* of a set of rules, each of which has exactly the same support and confidence, and it can be easily reconstructed from the representative. For example, assume we have the following association rules: $r_1 = i_1 \rightarrow i_2 \wedge i_3 \wedge i_4$, $r_2 = i_1 \wedge i_2 \rightarrow i_3 \wedge i_4$ and $r_3 = i_1 \wedge i_2 \wedge i_3 \rightarrow i_4$, where their supports and confidences are exactly identical. Given r_1, the others can be reconstructed from r_1 by a quite simple operation. Furthermore, their precise supports and confidences can be obtained immediately. In this sense, r_2 and r_3 are considered to be redundant and r_1 to be non-redundant [3]. Identifying non-redundant rules is just sufficient to obtain the possible ones. As has been mentioned above, since a non-redundant rule corresponds to a representative of a set of rules, the number of non-redundant rules is expected to be much smaller than one of the possible rules. By considering only non-redundant rules, therefore, we can drastically reduce the number of rules to be generated.

From the author's viewpoint, however, there often still exist many non-redundant rules. It might be a costly task for users to check them. Although we can easily reconstruct any redundant rule from a non-redundant one with its

[2] If a rule meets the requirements, we say that the rule is *significant*.

[3] In a word, such a non-redundant rule is characterized as one with the minimal antecedent and the maximal consequent.

precise support and confidence according to the original framework, the authors would like to claim that

> from a practical point of view, even though we cannot surely derive the precise support and confidence of redundant rule, it would be worth reducing the number of output rules further.

We try in this paper to propose a method for such a reduction by extending the original approach. Especially for this purpose, the original notion of redundancy is extended according to the claim above.

Since the support and confidence of our redundant rule can be *approximately* derived from a non-redundant one according to such an extended redundancy, these approximate values might not satisfy some users who require a high precision of the derived values. In our framework, therefore, we can flexibly adjust the maximum error by giving an adequate value of a user-defined parameter ε ($0 \leq \varepsilon < 1$). As ε approaches 1, the maximum error increases, but the number of non-redundant rules decreases. Conversely, as ε approaches 0, the maximum error approaches 0, but the number of non-redundant rules increases.

Given a user-defined parameter ε, in order to describe our non-redundancy, we define a set of rules w.r.t. ε, called an *approximate informative basis* ($AIB(\varepsilon)$). It will be proved that every rule r in $AIB(\varepsilon)$ has the following property:

> Any rule r' reconstructable from r has *approximately* the same support and confidence as ones of r, where the maximum errors of these values are evaluated by some formulas determined by ε.

Thus a rule reconstructable from r is redundant. For the same reason, such a rule r in $AIB(\varepsilon)$ is non-redundant, and can approximately represent any rule reconstructable from it.

For any significant rule r, there always exists a corresponding non-redundant rule in $AIB(\varepsilon)$ from which r can be reconstructed. No significant rule can be lost, once $AIB(\varepsilon)$ is computed. The *completeness* in this sense and *weak-soundness* of $AIB(\varepsilon)$ are summarized in a theorem. We present an algorithm for constructing $AIB(\varepsilon)$. An effectiveness of our method is shown by some experimental results.

This paper is organized as follows. In the next section, we introduce some terminologies used throughout this paper. In Section 3, we briefly explain the original framework by Bastide, et al. Section 4 discusses our method for constructing $AIB(\varepsilon)$ with an example. Our preliminary experimental results are presented in Section 5. We summarize this paper and give some discussions in the last section. Especially, we briefly describe a new interactive strategy, which we are going to develop, for identifying interesting rules based on the method presented in this paper.

2 Preliminaries

Let $\mathcal{I}$ be a finite set of *items*. An *itemset* l is a non-empty subset of $\mathcal{I}$. A tuple $\langle id, l \rangle$ is called a *transaction*, where id is a transaction identifier and l is an

itemset. A *transaction database* $\mathcal{D}$ is a finite set of transactions. We often refer to $itemset(id)$ as the itemset associated with id in a transaction.

For a transaction $t = \langle id, l \rangle$, we say that t *contains* an itemset l' if $l' \subseteq l$. Given a transaction database $\mathcal{D}$, the *support of an itemset* l, denoted by $sup(l)$, is defined as the ratio of the number of transactions containing l to the number of all transactions in $\mathcal{D}$. Let $minsup$ be a user-defined threshold for the permissive minimum support. An itemset l is called a *frequent* itemset if $sup(l) \geq minsup$.

An *association rule* r is an implication between two itemsets which is of the form $r = l_1 \rightarrow (l_2 \setminus l_1)$, where l_1 and l_2 are itemsets such that $l_1 \subset l_2$. The *support of* r, denoted by $sup(r)$, is defined as $sup(r) = sup(l_2)$. Furthermore, the *confidence of* r, denoted by $conf(r)$, is defined as $conf(r) = sup(l_2)\ /\ sup(l_1)$. Let $minconf$ be a user-defined threshold for the permissive minimum confidence. An association rule r is said to be *significant* if $sup(r) \geq minsup$ and $conf(r) \geq minconf$.

Given a transaction database $\mathcal{D}$, let $\mathcal{ID}$ be the set of transaction identifiers in $\mathcal{D}$. We consider a mapping $\psi : 2^{\mathcal{I}} \rightarrow 2^{\mathcal{ID}}$ that is defined as $\psi(l) = \{id \mid \langle id, l' \rangle \in \mathcal{D} \wedge l \subseteq l'\}$. Moreover, we consider a mapping $\varphi : 2^{\mathcal{ID}} \rightarrow 2^{\mathcal{I}}$ that is defined as $\varphi(ID) = \bigcap_{id \in ID} itemset(id)$. Based on these mappings, a *closure operator* $\gamma : 2^{\mathcal{I}} \rightarrow 2^{\mathcal{I}}$ is defined as $\gamma(l) = \varphi(\psi(l))$, that is, γ computes the maximum itemset that is shared with all transactions containing l.

We say that an itemset l is *closed* if $\gamma(l) = l$. Since $\gamma(\gamma(l)) = \gamma(l)$ holds for any itemset l, $\gamma(l)$ is a closed itemset. It should be noted that for any itemset l' such that $l \subseteq l' \subseteq \gamma(l)$, $\gamma(l') = \gamma(l)$ and $sup(l) = sup(l') = sup(\gamma(l))$ hold.

An itemset l is called an *exact generator* (*E-generator*) of $\gamma(l)$. For a frequent closed itemset f, we refer to the set of E-generators of f as $EG(f)$ and the set of *minimal E-generators* of f as $MEG(f)$, that is, $MEG(f) = \{\ g\ \mid\ g \in EG(f) \wedge\ \nexists g' \in EG(f)$ such that $g' \subset g\ \}$. For a frequent closed itemset f and its E-generator $g \in MEG(f)$, a tuple (g, f) is called an *EGC-tuple*. Given an EGC-tuple (g, f), for any itemset l such that $g \subseteq l \subseteq f$, $sup(g) = sup(l) = sup(f)$ holds. The set of EGC-tuples w.r.t. $\mathcal{D}$ is referred to as $EGC(\mathcal{D})$.

3 Informative Basis of Association Rules

In this section, we briefly introduce a method of reducing the number of generated rules [2]. The key notion of this approach is a *redundancy* of association rule.

Definition 1. (Redundancy of Association Rule) [2]
Let $r = l_1 \rightarrow (l_2 \setminus l_1)$ be an association rule. r is called a *redundant* rule iff there exists an association rule $r' = l'_1 \rightarrow (l'_2 \setminus l'_1)$ such that $l'_1 \subseteq l_1$, $l_2 \subseteq l'_2$, $r' \neq r$, $sup(r') = sup(r)$ and $conf(r') = conf(r)$. ∎

Intuitively speaking, a redundant rule r is a rule which has *exactly* the same support and confidence as ones of some non-redundant rule r' and can be easily *reconstructed* from r' by a simple operation on itemsets. Therefore, a non-redundant rule can be viewed as a *representative* of a set of redundant ones. This

implies that extracting only non-redundant rules can be considered sufficient for the discovery of all possible rules. Since it is obvious that the number of non-redundant rules is smaller than that of all rules, we can reduce the number of rules to be obtained by simply taking non-redundant ones into account.

Each non-redundant rule is characterized as a rule with the minimal antecedent and maximal consequent and is formally defined in terms of E-generator and closure. It is shown that any rule can be reconstructed from a non-redundant rule with its precise support and confidence.

Furthermore, some experimental results show that the number of non-redundant rules is much smaller than that of all possible rules. Therefore, the method can be considered effective and promising in order to reduce the number of rules to be generated. However, there often exist a large number of non-redundant rules even though all redundant ones are discarded. Since the task of checking them would be still costly for users, more reduction is strongly desired to assist the user's task.

In the next section, we try to propose a method for such a reduction by extending the original approach.

4 Approximate Informative Basis of Association Rules

As just mentioned, we still have a large number of rules even if we consider redundant ones to be unnecessary. Although we can easily reconstruct any redundant rule from a non-redundant one with its precise support and confidence according to the original framework, the authors would like to claim that

> from a practical point of view, even though we cannot precisely derive the supports and confidences of redundant rules, it would be worth reducing the number of output rules further.

In this section, we try to propose a method for reducing the number of rules to be generated. Especially for this purpose, the original notion of redundancy is extended according to the claim above. In order to present our method, we first introduce a notion of *approximate generator*.

4.1 Approximate Generators of Closed Itemsets

An *approximate generator* is an extension of E-generator and it can work more flexibly.

Definition 2. **(A-Generators)**
Let l be an itemset and f a closed itemset. l is called an *approximate generator* (*A-generator*) of f if $\gamma(l) \subseteq f$ and $sup(f)$ / $sup(\gamma(l)) \geq 1 - \varepsilon$, where ε is a user-defined parameter ($0 \leq \varepsilon < 1$). ∎

> Note that any E-generator of a closed itemset f is an A-generator of f.
> The following property plays a very important role in our method.

Proposition 1.
Let g be an A-generator of a closed itemset f. For any itemset l such that $g \subseteq l \subseteq f$,

$$sup(g) \geq sup(l) \geq (1-\varepsilon)sup(g)$$

and

$$sup(f) \ / \ (1-\varepsilon) \geq sup(l) \geq sup(f).$$

Proof.
From the definition of A-generator, $1 \geq sup(f) \ / \ sup(\gamma(g)) \geq 1-\varepsilon$ holds. Since $sup(\gamma(g)) = sup(g)$, we have $sup(g) \geq sup(f) \geq (1-\varepsilon)sup(g)$. From $sup(g) \geq sup(l) \geq sup(f)$, therefore, $sup(g) \geq sup(l) \geq (1-\varepsilon)sup(g)$ holds.

Based on the inequalities above, we can easily obtain $sup(f) \ / \ (1-\varepsilon) \geq sup(l) \geq sup(f)$ as well. □

The proposition implies that $sup(g)$ and $sup(f)$ can be considered as *approximations* of $sup(l)$ if we could accept the errors. It should be noted here that the maximum errors are precisely evaluated with the parameter ε. Therefore, we can flexibly adjust the maximum errors so that they are permissible for us. As ε approaches 1, the maximum becomes larger. Conversely, as ε approaches 0, the maximum error approaches 0. That is, in case of $\varepsilon = 0$, any A-generator corresponds to an E-generator.

4.2 Approximation of *EGC*-Tuples

As previously mentioned, for any itemset l, the support of l can be *precisely* identified with an EGC-tuple (g, f) such that $g \subseteq l \subseteq f$, since $sup(g) = sup(l) = sup(f)$. Therefore, based on the set of EGC-tuples w.r.t. $\mathcal{D}$, $EGC(\mathcal{D})$, we can obtain the precise support of any itemset.

On the other hand, we define here an *approximation of* $EGC(\mathcal{D})$ with the help of A-generators. Using the approximation, we can *approximately* identify the support of any itemset with the maximum errors we just discussed.

Definition 3. (Approximation of $EGC(\mathcal{D})$)
Let $\mathcal{F}$ be the set of frequent closed itemsets w.r.t. $\mathcal{D}$ and ε, a user-defined parameter ($0 \leq \varepsilon < 1$). Consider a partition of $\mathcal{F}$, $\{F_1, \ldots, F_k\}$ [4]. For each F_i, there uniquely exists a closure $f_i^* \in F_i$ such that $\forall f \in F_i \ f \subseteq f_i^*$ and $sup(f_i^*)/sup(f) \geq 1-\varepsilon$. For each F_i, let us consider $AGC(F_i) = \{(g, f_i^*) \mid g \in min(\bigcup_{f \in F_i} MEG(f))\}$ [5]. An *approximation of* $EGC(\mathcal{D})$ is defined as

$$AGC(\mathcal{D}, \varepsilon) = \bigcup_{i=1}^{k} AGC(F_i).$$

Each tuple in $AGC(\mathcal{D}, \varepsilon)$ is called an *AGC-tuple*. ■

[4] That is, $\mathcal{F} = \cup_{i=1}^{k} F_i$ and $F_i \cap F_j = \phi$ $(i \neq j)$, where each F_i is called a *cell*.
[5] For a set S, $min(S)$ denotes the set of minimal elements in S under the set-inclusion ordering.

From the definition, for each EGC-tuple $(g, f) \in EGC(\mathcal{D})$, it is obvious that f uniquely belongs to some F_i and there exists an AGC-tuple $(g^*, f_i^*) \in AGC(\mathcal{D}, \varepsilon)$ such that $g^* \subseteq g$ and $f \subseteq f_i^*$. Moreover, for any AGC-tuple (g^*, f^*), g^* is an A-generator of f^*. From these observations and Proposition 1, therefore, we can obtain the following statement.

Proposition 2.
For any frequent itemset l, there exists an AGC-tuple $(g, f) \in AGC(\mathcal{D}, \varepsilon)$ such that $g \subseteq l \subseteq f$. Furthermore, $sup(g) \geq sup(l) \geq (1-\varepsilon)sup(g)$ and $sup(f) \ / \ (1-\varepsilon) \geq sup(l) \geq sup(f)$ hold.

Proposition 2 implies that $AGC(\mathcal{D}, \varepsilon)$ can identify the support of any frequent itemset *approximately*, where the maximum errors are precisely evaluated by functions of ε.

4.3 Approximate Informative Basis of Association Rules

Based on the set of AGC-tuples, $AGC(\mathcal{D}, \varepsilon)$, we can construct a basis of association rules, called an *approximate informative basis*(AIB), from which any significant rule can be easily reconstructed with its approximate support and confidence. Before giving the formal definition, we introduce a notion of *approximate source* of association rules.

Definition 4. (Approximate Sources of Association Rules)
Let $\mathcal{D}$ be a transaction database, ε a user-defined parameter ($0 \leq \varepsilon < 1$) and $\mathcal{F}$ the set of frequent closed itemsets. Assume that $\{F_1, \ldots, F_k\}$ is the partition of $\mathcal{F}$ based on which $AGC(\mathcal{D}, \varepsilon)$ is constructed.

For an EGC-tuple $(g, f) \in EGC(\mathcal{D})$, consider an F_i such that $f \subseteq f_i^*$. An association rule to which the pair of (g, f) and $AGC(F_i)$ is attached,

$$s = g \rightarrow (f_i^* \setminus g) \ : \ \langle \ (g, f) \ , \ AGC(F_i) \ \rangle,$$

is called an *approximate source* (A-source) of association rules [6]. The set of A-sources is referred to as $AS(\mathcal{D}, \varepsilon)$. ∎

We can reconstruct a set of association rules from an A-source.

Definition 5. (Reconstruction of Association Rules from A-source)
Let $s = g \rightarrow (f^* \setminus g) \ : \ \langle \ (g, f) \ , \ AGC(F) \ \rangle$ be an A-source. It is said that an association rule $l_1 \rightarrow (l_2 \setminus l_1)$ can be *reconstructed from* s if $g \subseteq l_1 \subseteq f$ and for an AGC-tuple $(g^*, f^*) \in AGC(F)$, $g^* \subseteq l_2 \subseteq f^*$. ∎

As shown in the next proposition, for any association rule that is reconstructed from an A-source, its support and confidence can be within certain ranges determined by the values of the source and ε.

[6] In what follows, depending on contexts, s often denotes only the rule $g \rightarrow (f_i^* \setminus g)$ of s.

Proposition 3.
Let s be an A-source and r be an association rule reconstructed from s. Then

$$\frac{sup(s)}{1-\varepsilon} \geq sup(r) \geq sup(s) \qquad \text{and} \qquad \frac{conf(s)}{1-\varepsilon} \geq conf(r) \geq conf(s)$$

hold.

Proof.
Let $s = g \rightarrow (f^* \setminus g) \ : \ \langle \ (g, f) \ , \ AGC(F) \ \rangle$ be an A-source and $r = l_1 \rightarrow (l_2 \setminus l_1)$ be an association rule reconstructed from s. From the definition of reconstruction, $g \subseteq l_1 \subseteq f$ and for an AGC-tuple (g^*, f^*) in $AGC(F)$, $g^* \subseteq l_2 \subseteq f^*$ hold. Note here that $sup(g) = sup(l_1) = sup(f)$. Furthermore, from Proposition 1, $sup(g^*) \geq sup(l_2) \geq (1-\varepsilon)sup(g^*)$ and $sup(f^*)/(1-\varepsilon) \geq sup(l_2) \geq sup(f^*)$ holds.

Since $sup(s) = sup(f^*)$ and $sup(r) = sup(l_2)$, we can immediately obtain $sup(s)/(1-\varepsilon) \geq sup(r) \geq sup(s)$.

Moreover, since $sup(g) = sup(l_1)$ and $sup(l_2) \geq sup(f^*)$, $sup(l_2)/sup(l_1) \geq sup(f^*)/sup(g)$ holds. Similarly, from $sup(g) = sup(l_1)$ and $sup(f^*)/(1-\varepsilon) \geq sup(l_2)$, $sup(f^*)/\{(1-\varepsilon)sup(g)\} \geq sup(l_2)/sup(l_1)$ holds. Therefore, we obtain $sup(f^*)/\{(1-\varepsilon)sup(g)\} \geq sup(l_2)/sup(l_1) \geq sup(f^*)/sup(g)$, that is, $conf(s)/(1-\varepsilon) \geq conf(r) \geq conf(s)$. □

The proposition states that if we could accept the errors, then $sup(s)$ and $conf(s)$ can be viewed as approximations of $sup(r)$ and $conf(r)$, respectively. That is, a set of association rules can be easily reconstructed from an A-source with their approximate supports and confidences. In this sense, we can consider these rules to be *approximately redundant* (*A-redundant*).

Now we can define an *approximate informative basis* of association rules from which any significant rule can be reconstructed with its approximate values of support and confidence.

Definition 6. (Approximate Informative Basis of Association Rules)
Let $\mathcal{D}$ be a transaction database, ε be a user-defined parameter ($0 \leq \varepsilon < 1$). An *approximate informative basis* of the significant association rules w.r.t. $\mathcal{D}$ and ε, denoted by $AIB(\mathcal{D}, \varepsilon)$, is defined as the set of A-sources whose confidences are not less than $(1-\varepsilon)minconf$:

$$AIB(\mathcal{D}, \varepsilon) = \{ \ s \mid s \in AS(\mathcal{D}, \varepsilon) \wedge conf(s) \geq (1-\varepsilon)minconf \ \}.$$

■

Theorem 1.

Weak-Soundness of $AIB(\mathcal{D}, \varepsilon)$:
Any association rule r reconstructed from s in $AIB(\mathcal{D}, \varepsilon)$ is significant or at worst A-significant [7].

[7] For an association rule r, if $sup(r) \geq minsup$ and $minconf > conf(r) \geq (1-\varepsilon)minconf$, we say that r is approximately significant (A-significant).

Completeness of $AIB(\mathcal{D}, \varepsilon)$:
For any significant association rule r, there exists an A-source s in $AIB(\mathcal{D}, \varepsilon)$ from which r can be reconstructed.

Proof.
Weak-Soundness: Let $r = l_1 \rightarrow (l_2 \setminus l_1)$ be an association rule reconstructed from an A-source $s = g \rightarrow (f^* \setminus g) \ : \ \langle \ (g, f) \ , \ AGC(F) \ \rangle$ in $AIB(\mathcal{D}, \varepsilon)$. Then, there exists an AGC-tuple (g^*, f^*) in $AGC(F)$ such that $g^* \subseteq l_2 \subseteq f^*$.

From Proposition 3, $sup(r) \geq sup(s)$ and $conf(r) \geq conf(s)$. Since f^* is a frequent closed itemset, $sup(f^*) \geq minsup$. From $sup(s) = sup(f^*)$, therefore, we have $sup(r) \geq minsup$. Furthermore, since $conf(s) \geq (1-\varepsilon)minconf$, we immediately have $conf(r) \geq (1-\varepsilon)minconf$. Therefore, r is at worst A-significant.
Completeness: Let $r = l_1 \rightarrow (l_2 \setminus l_1)$ be a significant association rule. For each l_i, there exists an EGC-tuple (g_i, f_i) in $EGC(\mathcal{D})$ such that $g_i \subseteq l_i \subseteq f_i$. It should be noted here that since $l_1 \subset l_2$, $f_1 \subseteq f_2$ holds. Assume that $AGC(\mathcal{D}, \varepsilon)$ is constructed based on a partition of $\mathcal{F}$, $\mathcal{P}_{\mathcal{F}}$. For the EGC-tuple (g_2, f_2), we can consider a cell F of $\mathcal{P}_{\mathcal{F}}$ such that $f_2 \subseteq f^*$, where f^* is the maximum itemset in F. Therefore, there exists an AGC-tuple (g^*, f^*) in $AGC(F)$ such that $g^* \subseteq g_2 \subseteq f_2 \subseteq f^*$. Furthermore, $f_1 \subseteq f^*$ holds. Therefore, $s = g_1 \rightarrow (f^* \setminus g_1) \ : \ \langle \ (g_1, f_1) \ , \ AGC(F) \ \rangle$ is an A-source from which r can be reconstructed.

Since r is a significant rule, $sup(l_2)/sup(l_1) \geq minconf$ holds. By multiplying both sides by $(1-\varepsilon)$, we obtain $(1-\varepsilon)sup(l_2)/sup(l_1) \geq (1-\varepsilon)minconf$. From Proposition 2, $sup(f^*)/(1-\varepsilon) \geq sup(l_2)$ holds, that is, $sup(f^*) \geq (1-\varepsilon)sup(l_2)$. Therefore, we have $sup(f^*)/sup(l_1) \geq (1-\varepsilon)minconf$. Since $sup(l_1) = sup(g_1)$ and $sup(f^*)/sup(g_1) = conf(s)$, $AIB(\mathcal{D}, \varepsilon)$ contains the A-source s. □

From Theorem 1, it is ensured that once we have $AIB(\mathcal{D}, \varepsilon)$, no significant rule can be lost.

4.4 Constructing Approximate Informative Basis

Given a transaction database $\mathcal{D}$, $minsup$, $minconf$ and a user-defined parameter ε, we can construct an approximate informative basis w.r.t. $\mathcal{D}$ and ε, $AIB(\mathcal{D}, \varepsilon)$. The construction process is divided into three sub-tasks:

1. Computing the set of EGC-tuples, $EGC(\mathcal{D})$.
2. Computing an approximation of $EGC(\mathcal{D})$, $AGC(\mathcal{D}, \varepsilon)$.
3. Constructing an approximate informative basis, $AIB(\mathcal{D}, \varepsilon)$.

The first task can be performed by adopting a Close [3]-like algorithm and the last one is straightforward. An algorithm for the second task, computing $AGC(\mathcal{D}, \varepsilon)$ from $EGC(\mathcal{D})$, is shown in Figure 1. In general, as ε becomes larger, the number of iteration for the while-loops decreases. The worst case complexity of the algorithm is $O(N^2)$, where N is the size of $EGC(\mathcal{D})$ (that is, the number of EGC-tuples in $EGC(\mathcal{D})$).

Input : $EGC(\mathcal{D})$ and ε.
Output : $AGC(\mathcal{D}, \varepsilon)$.
$AGC(\mathcal{D}, \varepsilon) \leftarrow \phi$;
$EG \leftarrow \phi$; $Rem \leftarrow \phi$; $Min \leftarrow \phi$;
while $EGC(\mathcal{D}) \neq \phi$ **do**
  pick up $t = (g, f)$ from $EGC(\mathcal{D})$;
  while $EGC(\mathcal{D}) \neq \phi$ **do**
    remove $t' = (g', f')$ from $EGC(\mathcal{D})$;
    If $f' \subseteq f \quad \wedge \quad sup(f)/sup(f') \geq 1 - \varepsilon$
      then $EG \leftarrow EG \cup \{g'\}$;
      else $Rem \leftarrow Rem \cup \{t'\}$;
    end
  end
  $Min \leftarrow$ the set of minimal elements of EG;
  for $g \in Min$ **do**
    $AGC(\mathcal{D}, \varepsilon) \leftarrow AGC(\mathcal{D}, \varepsilon) \cup \{(g, f)\}$;
  end
  $EGC(\mathcal{D}) \leftarrow Rem$;
  $EG \leftarrow \phi$; $Rem \leftarrow \phi$; $Min \leftarrow \phi$;
end
Output $AGC(\mathcal{D}, \varepsilon)$

Fig. 1. Algorithm for Constructing $AGC(\mathcal{D}, \varepsilon)$

Example:

For the transaction database $\mathcal{D}$ shown in Figure 2, we try to construct an approximate informative basis. In the database, each itemset is represented in a simple form. For example, an itemset $\{a, b, c\}$ is denoted as abc. We assume here that $minsup = 1/6$ and $\varepsilon = 0.7$.

At first, the set of EGC-tuples, $EGC(\mathcal{D})$, is computed. For the database, we can obtain the following 10 EGC-tuples:

$$\begin{aligned} EGC(\mathcal{D}) = \{\ & (a, ac) : 3/6,\ (b, b) : 5/6,\ (c, c) : 5/6,\ (d, acd) : 2/6,\ (e, be) : 4/6, \\ & (ab, abc) : 2/6,\ (ae, abce) : 1/6,\ (bc, bc) : 4/6, \\ & (bd, abcd) : 1/6,\ (ce, bce) : 3/6\ \}, \end{aligned}$$

where the value attached to each tuple is the support of the tuple.

Then an $AGC(\mathcal{D}, \varepsilon)$ is constructed from $EGC(\mathcal{D})$ according to the algorithm in Figure 1. For example, we have

$$AGC(\mathcal{D}, \varepsilon) = \{\ (a, abce),\ (ce, abce),\ (d, abcd),\ (b, be),\ (e, be),\ (c, bc)\ \}.$$

It should be noted here that the set of frequent closed itemsets,

$$\mathcal{F} = \{\ abce,\ abcd,\ abc,\ bce,\ acd,\ ac,\ be,\ bc,\ b,\ c\ \},$$

is divided into the following 4 cells:

ID	itemset
1	acd
2	bce
3	abce
4	be
5	abcd
6	bce

Fig. 2. Example of Transaction Database

$$
\begin{aligned}
&F_1 = \{\ abce,\ abc,\ bce,\ ac\ \},\\
&F_2 = \{\ abcd,\ acd\ \},\\
&F_3 = \{\ be,\ b\ \} \quad \text{and}\\
&F_4 = \{\ bc,\ c\ \}.
\end{aligned}
$$

That is,

$$
\begin{aligned}
&AGC(F_1) = \{\ (a, abce),\ (ce, abce)\ \},\\
&AGC(F_2) = \{\ (d, abcd)\ \},\\
&AGC(F_3) = \{\ (b, be),\ (e, be)\ \} \quad \text{and}\\
&AGC(F_4) = \{\ (c, bc)\ \}.
\end{aligned}
$$

Based on $AGC(\mathcal{D}, \varepsilon)$, we can obtain the set of A-sources, $AS(\mathcal{D}, \varepsilon)$, consisting of 20 sources. Assuming $minconf = 0.85$, we have the following approximate informative basis consisting of 12 sources:

$$
\begin{aligned}
AIB(\mathcal{D}, \varepsilon) = \{\ s_1 &= a \rightarrow (abce \setminus a)\ :\ \langle (a, ac), AGC(F_1)\rangle,\\
s_2 &= a \rightarrow (abcd \setminus a)\ :\ \langle (a, ac), AGC(F_2)\rangle,\\
s_3 &= b \rightarrow (be \setminus b)\ :\ \langle (b, b), AGC(F_3)\rangle,\\
s_4 &= b \rightarrow (bc \setminus b)\ :\ \langle (b, b), AGC(F_4)\rangle,\\
s_5 &= c \rightarrow (bc \setminus c)\ :\ \langle (c, c), AGC(F_4)\rangle,\\
s_6 &= d \rightarrow (abcd \setminus d)\ :\ \langle (d, acd), AGC(F_2)\rangle,\\
s_7 &= e \rightarrow (be \setminus e)\ :\ \langle (e, be), AGC(F_3)\rangle,\\
s_8 &= ab \rightarrow (abce \setminus ab)\ :\ \langle (ab, abc), AGC(F_1)\rangle,\\
s_9 &= ab \rightarrow (abcd \setminus ab)\ :\ \langle (ab, abc), AGC(F_2)\rangle,\\
s_{10} &= ae \rightarrow (abce \setminus ae)\ :\ \langle (ae, abce), AGC(F_1)\rangle,\\
s_{11} &= bd \rightarrow (abcd \setminus bd)\ :\ \langle (bd, abcd), AGC(F_2)\rangle,\\
s_{12} &= ce \rightarrow (abce \setminus ce)\ :\ \langle (ce, bce), AGC(F_1)\rangle\ \}.
\end{aligned}
$$

Table 1. Experimental Results

$minsup = 0.1$

	$minconf = 0.7$	$minconf = 0.5$	$minconf = 0.3$
Close	5,134	9,290	15,048
Our System ($\varepsilon = 0.1$)	1,733	2,985	4,444
Our System ($\varepsilon = 0.2$)	1,196	1,793	2,502

$minsup = 0.05$

	$minconf = 0.7$	$minconf = 0.5$	$minconf = 0.3$
Close	7,742	15,594	28,712
Our System ($\varepsilon = 0.1$)	3,203	5,817	9,822
Our System ($\varepsilon = 0.2$)	2,194	3,600	5,500

$minsup = 0.01$

	$minconf = 0.7$	$minconf = 0.5$	$minconf = 0.3$
Close	11,997	28,458	59,153
Our System ($\varepsilon = 0.1$)	6,900	13,290	25,113
Our System ($\varepsilon = 0.2$)	3,824	6,357	10,432

For example, from the A-source s_1, an association rule $r = a \rightarrow (ac \setminus a)$ can be reconstructed with its approximate support and confidence, $1/6$ ($= sup(s_1)$) and $1/3$ ($= conf(s_1)$). On the other hand, its precise support and confidence are $1/6$ and 1, respectively. We can easily verify that the error of the confidence surely follows Proposition 3.

5 Experimental Results

In this section, we present our preliminary experimental results.

In order to verify an effectiveness of our method, we have implemented a system to compute an AIB based on the algorithms presented in the previous section. The algorithm Close has been implemented as well to compare with the original method by Bastide, et al. Our system and Close have been written in C and have been tested on a 400MHz PentiumII PC with 160MB memory.

For our experimentation, we have obtained "1984 United States Congressional Voting Records Database", a database from the UCI Repository [7]. It consists of 435 transactions and the number of possible items is 17. Our system has computed AIBs for the database in various settings of parameters, $minsup$, $minconf$ and ε.

The numbers of rules output by each system are summarized in Table 1, where the results obtained by the original method is referred to as Close.

For each parameter setting, our system has output fewer rules compared to that by Close. In the most effective case, about 70% reduction has been achieved

compared to Close[8]. Even in the worst case, about 43% reduction has been achieved. Therefore, we can consider that our method is very effective to reduce the number of generated rules.

6 Concluding Remarks

In this paper, we have presented a method for constructing an approximate informative basis (AIB) for significant association rules from which any significant rule can easily be reconstructed with its approximate support and confidence. The maximum errors of these values are precisely evaluated by some formulae determined by a user-defined parameter ε. Therefore, we can flexibly adjust the preciseness of these approximate values. Some experimental results have shown that our method can drastically reduce the number of rules to be generated compared to the original framework. Therefore, readability and understandability for the rules would be improved by providing an adequate value of ε.

As a next step of this study, we are planning to formalize a method for identifying actually interesting rules with their support and confidence in an *interactive* manner. In the initial stage, ε is given a value close to 1 by a user and we obtain a rough AIB for which we can easily and completely check the contents. By checking them, the user selects several A-sources from which some interesting rules seem to be reconstructed. Then the user decreases the value of ε to obtain a more precise AIB. It should be noted that the system presents only a part of the AIB which is relative to the A-sources previously selected by the user. Therefore, we can obtain a more precise AIB keeping the number of contents small. For the presented AIB, similar processes are iteratively performed until the user satisfactorily identifies interesting rules with their support and confidence. At each stage, since the system keeps the number of contents of presented AIB compact, the selection tasks by the user would not be costly. Therefore, such a system would be quite helpful for users who try to discover interesting rules easily.

In order to construct such an interactive system, we are expecting that the efficiency of computing AIB has to be improved more. Our AIB is currently computed by adopting an extended algorithm of Close [3]. Although Close can efficiently identify the set of frequent closed itemsets, several new algorithms for the same task have been proposed recently, e.g., A-Close [4], CHARM [6] and CLOSET [5]. By adopting these algorithms, the efficiency of computation of AIB would be improved.

References

1. R. Agrawal, R. Srikant: Fast Algorithms for Mining Association Rules, *Proc. of the 20th Int'l Conf. on Very Large Data Bases,* pp. 478–499, 1994.

[8] It has been reported that Close has achieved about 80 – 90% reductions compared to Apriori.

2. Y. Bastide, N. Psquier, R. Taouil, G. Stumme and L. Lakhal: Mining Minimal Non-Redundant Association Rule Using Frequent Closed Itemset Proc. of Int'l Conf. on Computational Logic–CL2000, LNAI 1861, pp.972-986, 2000
3. N. Pasquier, Y. Bastide, B. Rafik and L. Lakhal: Efficient Mining of Association Rules Using Closed Itemset Lattices, Information Systems, vol. 24, no. 1, pp.25–46, 1999
4. N. Pasquier, Y. Bastide, B. Rafik and L. Lakhal: Discovering Frequent Closed Itemsets for Association Rules, Proc. of ICDT, LNCS 1540, pp.398–416 1999
5. J. Pei, J. Han and R. Mao: CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets, Proc. of DMKD2000, 2000
6. M. J. Zaki and C. Hsiao: CHARM: An Efficient Algorithm for Closed Association Rule Mining, Technical Report 99-10, Computer Science, Rensselaer Polytechnic Institute, 1999
7. P.M. Marphy and D. W. Aha.: UCI Repository of machine learning databases, http://www.ics.uci.edu/mlearn/MLRepository.html, Univ. of California, Dept. of Information and Computer Science, 1994

Passage-Based Document Retrieval as a Tool for Text Mining with User's Information Needs

Koichi Kise[1,2], Markus Junker[1], Andreas Dengel[1], and Keinosuke Matsumoto[2]

[1] German Research Center for Artificial Intelligence (DFKI GmbH),
P.O.Box 2080, 67608 Kaiserslautern, Germany
{Koichi.Kise, Markus.Junker, Andreas.Dengel}@dfki.de
[2] Department of Computer and Systems Sciences,
Graduate School of Engineering,
Osaka Prefecture University
1-1 Gakuencho, Sakai, Osaka 599-8531, Japan
{kise, matsu}@cs.osakafu-u.ac.jp

Abstract. Document retrieval can be considered as a basic but important tool for text mining that is capable of taking a user's information need into account. However, document retrieval is a hard task if multi-topic lengthy documents have to be retrieved with a very short description (a few keywords) of the information need. In this paper, we focus on this problem which is typical in real world applications. We experimentally validate that passage-based document retrieval is advantageous in such circumstances as compared to conventional document retrieval. Passage-based document retrieval is a kind of document retrieval which takes into account only small fractions (passages) of documents to judge the document relevance to the information need. As a passage-based method, we employ the method based on density distributions of keywords. This is compared with the following three conventional methods for document retrieval: the vector space model, pseudo-feedback, and latent semantic indexing. Experimental results show that the passage-based method is superior to the conventional methods if long documents have to be retrieved by short queries.

1 Introduction

The growing number of electronic textual documents has created the need of intelligent access to the information implied by them. The goal of text mining is to discover novel nuggets of information from a huge collection of documents to fulfill the need in the ultimate sense [1]. The unstructured nature of documents, however, makes it difficult to realize the goal in a general way. The current state-of-the-art is to approach the goal by integrating the tools developed so far in other related research areas [2], though their functionality and/or domains of interest are still restricted. In order to take a step forward, it would be required both to devise a novel combination of the tools and to polish them up.

A typical scenario of text mining would be that (1) information extraction is utilized to obtain the information from documents, (2) data mining is applied

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 155–169, 2001.

to the extracted information to derive novel information. In this scenario, information of interest is fixed in the first stage of the processing. Another possibility is mining based on a user's ad-hoc information need. In this scenario, document retrieval is a tool applied at the first stage of processing to select documents analyzed at later stages. Although the research area of document retrieval has several decades of history, it is still not trivial to retrieve documents relevant to a user's need. Two major problems uncovered through the research activities are as follows:

Multi-topic documents. If a document is beyond the length of abstracts, it often contains several topics. Even though one of them is relevant to the user's need, the rest are not necessarily relevant. As a result, these irrelevant parts severely disturb the retrieval of documents.

Short queries. It is common that a user's information need is fed to a system as a set of query terms. However, it is not an easy task for a user to transform the need into query terms. From the analysis of Web search logs, for example, it is well-known that typical users issue quite short queries consisting of several terms. Such queries are too poor to retrieve documents appropriately.

In conventional document retrieval, the retrieval of multi-topic documents is a hard task since there is no way to avoid the influence of irrelevant parts of documents. In order to tackle this problem, some researchers have proposed a different way of retrieval called passage-based document retrieval [3,4,5]. In passage-based document retrieval, documents are retrieved based only on fractions (passages) of documents in order not to be disturbed by the irrelevant parts. It has been shown in the literature that passage-based document retrieval outperforms conventional document retrieval in processing long documents [5]. To handle passages as units of retrieval is advantageous to the application to text mining since it also gives a clue to extract relevant parts from the documents.

In this paper, we experimentally validate that, for the second (short queries) problem, passage-based document retrieval is also superior to conventional document retrieval. As a method of passage-based retrieval, we utilize a method based on "density distributions" [6]. This method segments documents into passages dynamically in response to a query. As conventional methods, we employ the following three [7,8]: the vector space model, pseudo-feedback and latent semantic indexing.

2 Conventional Document Retrieval

Let us begin with an overview of conventional document retrieval methods. The task of document retrieval is to retrieve documents relevant to a given query from a fixed set of documents or a document database. In a common way to deal with documents as well as queries, they are represented using a set of index terms (simply called terms from now on) by ignoring their positions in documents and queries. Terms are determined based on words of documents in the database. In the following, t_i $(1 \leq i \leq m)$ and d_j $(1 \leq j \leq n)$ represent a term and a

document in the database, respectively, where m is the number of terms and n is the number of documents.

2.1 Vector Space Model

The vector space model (VSM) [7,8] is the simplest retrieval model. In the VSM, a document d_j is represented as a m dimensional vector:

$$\boldsymbol{d}_j = (w_{1j}, ..., w_{mj})^T \ , \tag{1}$$

where T indicates the transpose, w_{ij} is a weight of a term t_i in a document d_j. A query q is likewise represented as

$$\boldsymbol{q} = (w_{1q}, ..., w_{mq})^T \ , \tag{2}$$

where w_{iq} is a weight of a term t_i in a query q.

So far, a variety of schemes for computing weights have been proposed. In this paper, we employ a standard scheme called "tf-idf" defined as follows:

$$w_{ij} = \mathrm{tf}_{ij} \cdot \mathrm{idf}_i \ , \tag{3}$$

where tf_{ij} is the weight calculated using the term frequency f_{ij} (the number of occurrences of a term t_i in a document d_j), and idf_i is the weight calculated using the inverse of the document frequency n_i (the number of documents which contain a term t_i). In computing tf_{ij} and idf_i, the raw frequency is usually dampened by a function. We utilize $\mathrm{tf}_{ij} = \sqrt{f_{ij}}$ and $\mathrm{idf}_i = \log(n/n_i)$ where n is the total number of documents. The weight w_{iq} is similarly defined as $w_{iq} = \sqrt{f_{iq}}$ where f_{iq} is the frequency of a term t_i in a query q.

The result of retrieval is represented as a list of documents ranked according to their similarity to the query. The similarity $\mathrm{sim}(\boldsymbol{d}_j, \boldsymbol{q})$ between a document d_j and a query q is measured by the cosine of the angle between $\boldsymbol{d}_j$ and $\boldsymbol{q}$:

$$\mathrm{sim}(\boldsymbol{d}_j, \boldsymbol{q}) = \frac{\boldsymbol{d}_j^T \boldsymbol{q}}{\|\boldsymbol{d}_j\| \ \|\boldsymbol{q}\|} \ . \tag{4}$$

where $\| \cdot \|$ is the Euclidean norm of a vector.

2.2 Pseudo-Feedback

A problem of the VSM is that a query is often too short to rank documents appropriately. To cope with this problem, it has been proposed to enrich an original query by expanding it with terms in documents.

A method called "pseudo-feedback" [8] is known as a way to obtain the terms for expansion. In this method, first, documents are ranked with an original query. Then, highly ranked documents are assumed to be relevant and their terms are incorporated into the original query. Documents are ranked again by using the expanded query.

In this paper, we employ a simple variant of pseudo-feedback. Let E be a set of document vectors for expansion given by

$$E = \left\{ \boldsymbol{d}_j^+ \middle| \frac{\mathrm{sim}(\boldsymbol{d}_j^+, \boldsymbol{q})}{\max_i \mathrm{sim}(\boldsymbol{d}_i, \boldsymbol{q})} \geq \tau \right\} \ , \tag{5}$$

where $\boldsymbol{q}$ is an original query vector and τ is a threshold of the similarity. The sum $\boldsymbol{d}_s$ of document vectors in E:

$$\boldsymbol{d}_s = \sum_{\boldsymbol{d}_j^+ \in E} \boldsymbol{d}_j^+ \tag{6}$$

can be considered as enriched information about the original query. Then, the expanded query vector $\boldsymbol{q}'$ is obtained by

$$\boldsymbol{q}' = \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|} + \lambda \frac{\boldsymbol{d}_s}{\|\boldsymbol{d}_s\|} \ , \tag{7}$$

where λ is a parameter for controlling the weight of the newly incorporated component. Finally, documents are ranked again according to the similarity $\mathrm{sim}(\boldsymbol{d}_j, \boldsymbol{q}')$ to the expanded query.

2.3 Latent Semantic Indexing

Latent semantic indexing (LSI) [7,8] is another well-known way to improve the VSM. Let D be a term-by-document matrix defined by

$$D = (\hat{\boldsymbol{d}}_1, ..., \hat{\boldsymbol{d}}_n) \ , \tag{8}$$

where $\hat{\boldsymbol{d}}_j = \boldsymbol{d}_j / \|\boldsymbol{d}_j\|$. By applying the singular value decomposition, D is decomposed into the product of three matrices:

$$D = USV^T \ , \tag{9}$$

where U and V are matrices of size $m \times r$ and $n \times r$ ($r = \mathrm{rank}(D)$), respectively, and $S = \mathrm{diag}(\sigma_1, ..., \sigma_r)$ is a diagonal matrix with singular values σ_i ($\sigma_i \geq \sigma_j$ if $i \leq j$). Each row vector in U (V) corresponds to a r-dimensional vector representing a term (document).

By keeping only the $k(< r)$ largest singular values in S along with the corresponding columns in U and V, D is approximated by

$$D_k = U_k S_k V_k^T \ , \tag{10}$$

where U_k, S_k and V_k are matrices of size $m \times k$, $k \times k$ and $n \times k$, respectively. This approximation allows us to uncover "latent" semantic relation among terms as well as documents.

The similarity between a document and a query is measured as follows. Let $\boldsymbol{v}_j = (v_{j1}, ..., v_{jk})$ be a row vector in $V_k = (v_{ji})$ $(1 \leq j \leq n, 1 \leq i \leq k)$. In the k-dimensional (approximated) space, a document d_j is represented as

$$\boldsymbol{d}_j^* = S_k \boldsymbol{v}_j^T \ . \tag{11}$$

An original query is also represented in the k-dimensional space as

$$\boldsymbol{q}^* = U_k^T \boldsymbol{q} \ . \tag{12}$$

Then the similarity is obtained by $\mathrm{sim}(\boldsymbol{d}_j^*, \boldsymbol{q}^*)$.

3 Passage-Based Document Retrieval

Passages used in passage-based methods can be classified into three types: discourse, semantic and window [3]. Discourse passages are defined based on discourse units such as sentences and paragraphs. Semantic passages are obtained by segmenting text at the points where the subject of text changes. Window passages are determined based on the number of terms.

In this paper, we employ a passage-based method with window passages called "density distributions"(DD). The density distribution was first introduced to locate the descriptions of a word [9] and applied to passage retrieval by some of the authors [6].

The fundamental idea of DD is that parts of documents which densely contain the terms in a query are relevant to it. Figure 1 shows an example of a density distribution. The horizontal axis indicates the positions of terms in a document. The distribution of query terms in the document is shown as spikes in the figure: their height indicates the weight of a term. The density distribution shown in the figure is obtained by smoothing the spikes with a window function. The details are as follows.

Let $a_j(l)$ $(1 \leq l \leq L_j)$ be a term at the position l in a document d_j where L_j is the length of a document d_j measured in terms. The weighted distribution $b_j(l)$ of terms in a query q is defined by

$$b_j(l) = \begin{cases} w_{iq} \cdot \mathrm{idf}_i & \text{if } a_j(l) = t_{iq} \ , \\ 0 & \text{otherwise} \ . \end{cases} \tag{13}$$

Smoothing of $b_j(l)$ enables us to obtain the density distribution $dd_j(l)$ for a document d_j:

$$dd_j(l) = \sum_{x=-W/2}^{W/2} f(x) b_j(l-x) \ , \tag{14}$$

where $f(x)$ is a window function with a window size W. We employ the Hanning window function defined by

$$f(x) = \begin{cases} \frac{1}{2}(1 + \cos 2\pi \frac{x}{W}) & \text{if } |x| \leq W/2 \ , \\ 0 & \text{otherwise} \ , \end{cases} \tag{15}$$

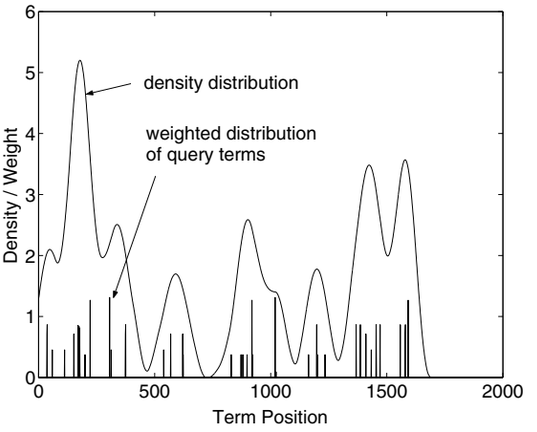


Fig. 1. Density distribution.

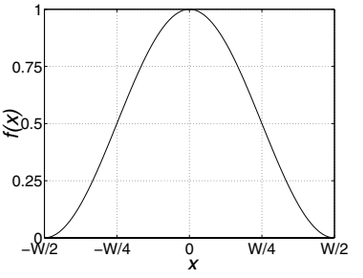


Fig. 2. Hanning window function.

whose shape is illustrated in Fig. 2.

In order to utilize DD as a passage-based document retrieval method, a score of a document is calculated using the density distribution. The score of d_j for a query q is obtained as the maximum value of its density distribution as follows:

$$\text{score}(d_j, q) = \max_l dd_j(l) \ . \tag{16}$$

This score is used to rank documents according to a query.

4 Experimental Comparison

In this section, we show the results of the experimental comparison. After the description of the test collections employed for the experiments, our methods for evaluating the results are described. Then, the results of experiments are presented and discussed.

Table 1. Statistics about documents in the test collections.

		MED	CRAN	CR	FR
size [MB]		1.1	1.6	235	209
no. of doc.		1,033	1,398	27,922	19,789
no. of terms†		4,284	2,550	37,769	43,760
doc. len.‡	min.	20	23	22	1
	max.	658	662	629,028	315,101
	mean	**155**	**162**	**1,455**	**1,792**
	median	139	142	324	550

† : counted in words *after* stemming and eliminating stopwords
‡ : counted in words *before* stemming and eliminating stopwords

Table 2. Statistics about queries in the test collections.

		MED	CRAN	CR			FR		
				title	desc	narr	title	desc	narr
no. of queries		30	225	34			85		
query len.†	min.	2	3	2	4	12	1	3	12
	max.	33	21	7	19	79	9	22	93
	mean	**10.8**	**9.2**	**3.0**	**7.7**	**28.7**	**3.5**	**10.4**	**37.0**
	median	9.0	9.0	3.0	6.5	24.5	3.0	10.0	34.0

† : counted in words *after* stemming and eliminating stopwords

4.1 Test Collections

We made a comparison using four test collections: MED (medicine), CRAN (aeronautics), FR (federal register), CR (congressional record). The collections MED and CRAN are available at [12], and FR and CR are contained in the TREC disks No.2 and No.4, respectively [13]. All collections are provided with queries and their groundtruth (a list of documents relevant to each query). For these collections, terms used for document representation were obtained by stemming and eliminating stopwords [1].

Tables 1 and 2 show some statistics about the collections. In Table 1, an important difference is the length of documents: MED and CRAN consist of abstracts, while FR and CR contain much longer documents. In Table 2, a point to note is the difference of query length. In the TREC collections, each information need is described by query types of different length. In order to investigate the influence of query length, we employed three types: "title"(the shortest representation), "desc" (description; medium length) and "narr" (narrative; the longest).

[1] Words which convey no meaning such as "the".

4.2 Evaluation

Average Precision. A common way to evaluate the performance of retrieval methods is to compute the (interpolated) precision at some recall levels. This results in a number of recall / precision points which are displayed in recall-precision graphs [7]. However, it is sometimes convenient for us to have a single value that summarizes the performance. *The average precision (non-interpolated) over all relevant documents* [7,12] is a measure resulting in a single value. The definition is as follows.

As described in Sect. 2, the result of retrieval is represented as the ranked list of documents. Let $r(i)$ be the rank of the i-th relevant document counted from the top of the list. The precision for this document is calculated by $i/r(i)$. The precision values for all documents relevant to a query are averaged to obtain a single value for the query. The average precision over all relevant documents is then obtained by averaging the respective values over all queries.

For example, consider two queries q_1 and q_2 which have two and three relevant documents, respectively. Suppose the ranks of relevant documents for q_1 are 2 and 5, and those for q_2 are 1, 3 and 10. The average precision for q_1 and q_2 is computed as $(1/2+2/5)/2 = 0.45$ and $(1/1+2/3+3/10)/3 = 0.66$, respectively. Then the average precision over all relevant documents which takes into account both queries is $(0.45 + 0.66)/2 = 0.56$.

Statistical Test. The next step for the evaluation is to compare the values of the average precision obtained by different methods. An important question here is whether the difference in the average precision is really meaningful or just by chance. In order to make such a distinction, it is necessary to apply a statistical test.

Several statistical tests have been applied to the task of information retrieval [10,11]. In this paper, we utilize the test called "macro t-test" [11] (called paired t-test in [10]). The following is the summary of the test as described in [10].

Let a_i and b_i be the scores (e.g., the average precision) of retrieval methods A and B for a query i and define $d_i = a_i - b_i$. The test can be applied under the assumptions that the model is additive, i.e., $d_i = \mu + \varepsilon_i$ where μ is the population mean and ε_i is an error, and that the errors are normally distributed. The null hypothesis here is $\mu = 0$ (A performs equivalently to B in terms of the average precision), and the alternative hypothesis is $\mu > 0$ (A performs better than B).

It is known that the Student's t-statistic

$$t = \frac{\bar{d}}{\sqrt{s^2/n}} \tag{17}$$

follows the t-distribution with the degree of freedom of $n - 1$, where n is the number of samples (queries), $\bar{d}$ and s^2 are the sample mean and variance:

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i \ , \tag{18}$$

Table 3. Values of parameters.

	parameter		MED, CRAN	CR, FR
PF	weight	λ	1.0, 2.0	1.0, 2.0
	threshold	τ	0.71 $\sim$ 0.99 step 0.02	0.71 $\sim$ 0.99 step 0.02
LSI	dimension	k	60 $\sim$ 500 step 20	50 $\sim$ 500 step 50
DD	window size	W	20 $\sim$ 200 step 20	20 $\sim$ 100 step 10, and 150,200,300

Table 4. Best parameter values.

		MED	CRAN	CR			FR		
				title	desc	narr	title	desc	narr
PF	λ	2.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0
	τ	0.71	0.85	0.85	0.85	0.93	0.83	0.71	0.71
LSI	k	60	260	300	500	400	350	500	500
DD	W	80	100	50	90	200	90	40	40

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(d_i - \bar{d})^2 \ . \tag{19}$$

By looking up the value of t in the t-distribution, we can obtain the P-value, i.e., the probability of observing the sample results d_i $(1 \leq i \leq n)$ under the assumption that the null hypothesis is true. The P-value is compared to a predetermined significance level α in order to decide whether the null hypothesis should be rejected or not. As significance levels, we utilize 0.05 and 0.01.

4.3 Results for the Whole Collections

The methods PF (pseudo-feedback), LSI (latent semantic indexing) and DD (density distributions) were applied by ranging the values of parameters as shown in Table 3. Figure 3 exemplarily illustrates the variation in the average precision when varying the threshold τ in PF ($\lambda = 1.0$; left) and the window size W in DD (right). The lines in the graphs were obtained from the experiments on the collections CR and FR. Since these collections have three query sets (title, desc, narr), six lines are shown in each graph. In the graph of PF, the average precision fluctuated slowly but irregularly with the threshold τ. On the other hand, the average precision of DD partly changed rapidly on smaller window sizes, and showed a tendency to converge as the window size became larger. Since better performance of DD was often obtained with smaller window sizes, DD would be more sensitive to the parameter W than PF to τ. Although it is an important topic to develop a method of automated adjustment of the window size, it is beyond the scope of this paper; we simply selected the best values of parameters which are shown in Table 4.

Table 5 shows the average precision obtained by using the best parameter values. In Table 5, the best and the second best values of average precision

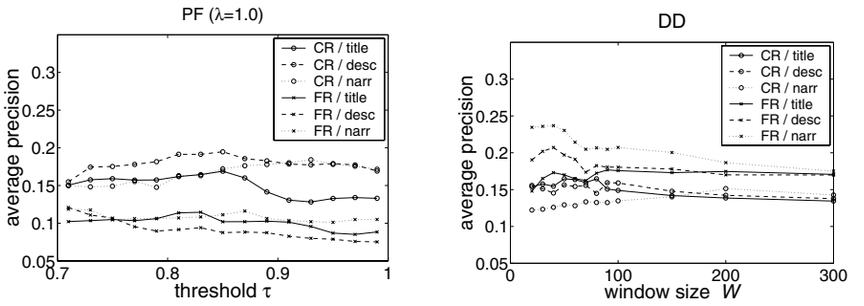


Fig. 3. Variations in the average precision.

Table 5. Average precision over all relevant documents.

	MED	CRAN	CR			FR		
			title	desc	narr	title	desc	narr
VSM	0.530	0.401	0.127	*0.172*	*0.172*	0.098	0.094	*0.120*
PF	*0.640*	**0.450**	**0.169**	**0.195**	**0.184**	*0.115*	*0.123*	0.119
	(+20.8%)	(+12.2%)	(+33.1%)	(+13.4%)	(+7.0%)	(+17.3%)	(+30.9%)	(−0.8%)
LSI	**0.685**	*0.444*	0.101	0.128	0.134	0.043	0.051	0.075
	(+29.2%)	(+10.7%)	(−20.5%)	(−25.6%)	(−22.1%)	(−56.1%)	(−45.7%)	(−37.5%)
DD	0.507	0.370	*0.165*	0.159	0.151	**0.177**	**0.207**	**0.237**
	(−4.3%)	(−7.7%)	(+29.9%)	(−7.6%)	(−12.2%)	(+80.6%)	(+120%)	(+97.5%)

( ) : difference to the VSM

among the methods are indicated in bold and italic fonts, respectively. In the parentheses, the ratio of difference to the VSM is noted. Let x and y be the average precision by the VSM and a method for comparison, respectively. The ratio is calculated by $(y - x)/x$. Thus a positive and a negative value indicate gain and loss, respectively.

The results of the macro t-test for all pairs of methods are shown in Table 6. The meaning of the symbols such as "$\gg$", "$>$" and "$\sim$" is summarized at the bottom of the table. For example, the symbol "$>$" was obtained in the case of DD compared to the VSM for the MED collection. This indicates that, at the significance level $\alpha = 0.05$, the null hypothesis "DD performs equivalently to the VSM" is rejected and the alternative hypothesis "DD performs worse than the VSM" is accepted. At $\alpha = 0.01$, however, the null hypothesis cannot be rejected. Roughly speaking, "$A \gg (\ll) B$", "$A > (<) B$" and "$A \sim B$" indicate that "A is almost guaranteed to be better (worse) than B", "A is likely to be better (worse) than B" and "A is equivalent to B", respectively.

The results shown in Tables 5 and 6 can be summarized as follows:

Table 6. Results of the macro t-test.

methods		MED	CRAN	CR			FR		
A	*B*			title	desc	narr	title	desc	narr
DD -	VSM	$<$	$\ll$	$\sim$	$\sim$	$\sim$	$\gg$	$\gg$	$\gg$
DD -	PF	$\ll$	$\ll$	$\sim$	$<$	$\sim$	$>$	$\gg$	$\gg$
DD -	LSI	$\ll$	$\ll$	$\sim$	$\sim$	$\sim$	$\gg$	$\gg$	$\gg$
PF -	VSM	$\gg$	$\gg$	$\gg$	$\sim$	$\sim$	$\sim$	$>$	$\sim$
PF -	LSI	$\ll$	$\sim$	$>$	$\sim$	$\sim$	$\gg$	$\gg$	$\gg$
LSI -	VSM	$\gg$	$\gg$	$\sim$	$\sim$	$\sim$	$\ll$	$\ll$	$\ll$

$\gg, \ll$: P-value ≤ 0.01
$>, <$: $0.01 <$ P-value ≤ 0.05
$\sim$: $0.05 <$ P-value

- For the collections of short documents (MED and CRAN), the methods PF and LSI outperformed the VSM and DD.
- For the collection CR which includes long documents, the methods mostly performed equivalently. The exception was the performance of PF. As shown in Table 6, PF was better than the VSM and LSI for the shortest queries (title) as well as DD for the middle length queries (desc).
  Note that methods are found to be equivalent by the statistical test even though the ratios of the difference of the average precision are bigger than those for MED and CRAN. For example, PF outperformed the VSM for MED and CRAN with the ratios +20.8% and +12.2%, while DD was equivalent to the VSM for CR with the ratio +29.9% (cmp. Table 5). This is because, in the statistical test, not only the average precision but also its variance and the number of queries are taken into account.
- For the collection FR which also includes long documents, on the other hand, DD clearly outperformed the other methods. The advantage of PF and LSI for the collections of short documents did not hold here.

From the above results, the influence of the length of documents and queries to the performance of the methods remains unclear. Although it has been shown that DD is inferior to PF and LSI for short documents, DD outperformed the other methods only for one of the collections which contain long documents. This could be because of the nature of the collections CR and FR. Although these collections include much longer documents than MED and CRAN, they also include many short documents as shown by the gap between the mean and the median in Table 1.

4.4 Results for Partitioned Collections

In order to clarify the relation between the performance and the length of documents and queries, we partitioned each of the collections CR and FR into three smaller collections as follows. Documents in the collections were first splitted

Table 7. Statistics about the partitioned collections.

	CR				FR			
	relevant doc.			irrel.	relevant doc.			irrel.
	short	middle	long	doc.	short	middle	long	doc.
no. of doc.	**251**	**251**	**252**	27,168	**148**	**148**	**148**	19,345
doc. len. min.	67	604	3,055	22	114	1,554	6,037	1
max.	601	3,029	629,028	385,065	1,512	5,994	315,101	124,353
mean	**334**	**1,315**	**33,550**	1,169	**859**	**3,075**	**35,982**	1,528
median	303	1,078	11,236	318	835	2,886	17,037	536
no. of queries	27	30	27	—	43	44	63	—

into two disjoint sets: documents relevant to at least one query, and those irrelevant to all queries. The set of relevant documents was further divided into three disjoint subsets of almost equal size according to the length of documents: short relevant documents, middle length relevant documents, and long relevant documents. By combining each subset with the set of irrelevant documents, we prepared three partitioned collections called "short", "middle" and "long". As queries for each partitioned collection, we took the queries which are relevant to at least one document in the partitioned collection. Since some documents are relevant to more than one query, the number of queries does not sum up to the number of queries in the original collections (cmp. Table 2). The statistics about the partitioned collections are shown in Table 7.

Using the best parameters as shown in Table 4, we computed the average precision for the partitioned collections. Figure 4 illustrates the results. Each graph in the figure represents the results for a pair of a set of partitioned collections and a query length. The horizontal axes of the graphs indicate the partitioned collections. These graphs show that the conventional methods (VSM, PF, LSI) performed worse as the documents became longer. On the other hand, DD yielded almost equivalent results for all document lengths on CR collection, and even better results for the FR collection as the documents were longer.

Table 8 shows the results of the statistical test for the partitioned collections. DD yielded significantly better results in most of the cases for the "long" partitions. These results confirm that passage-based document retrieval is better for longer documents, which has already been reported in the literature [5].

Let us now turn to the influence of the query length. Figure 5 illustrates the same results as in Fig. 4 but arranged in a different way. Here, each graph corresponds to a partitioned collection and the horizontal axes represent the query lengths.

For the "short" partitioned collections, no clear relation between the effectiveness of the methods and the query length could be found. On the other hand, for the "middle" and "long" partitioned collections with the shortest queries (title), DD was always the best among the methods. For the "middle" CR collection with longer queries, DD performed worse than the other methods. For the "long"

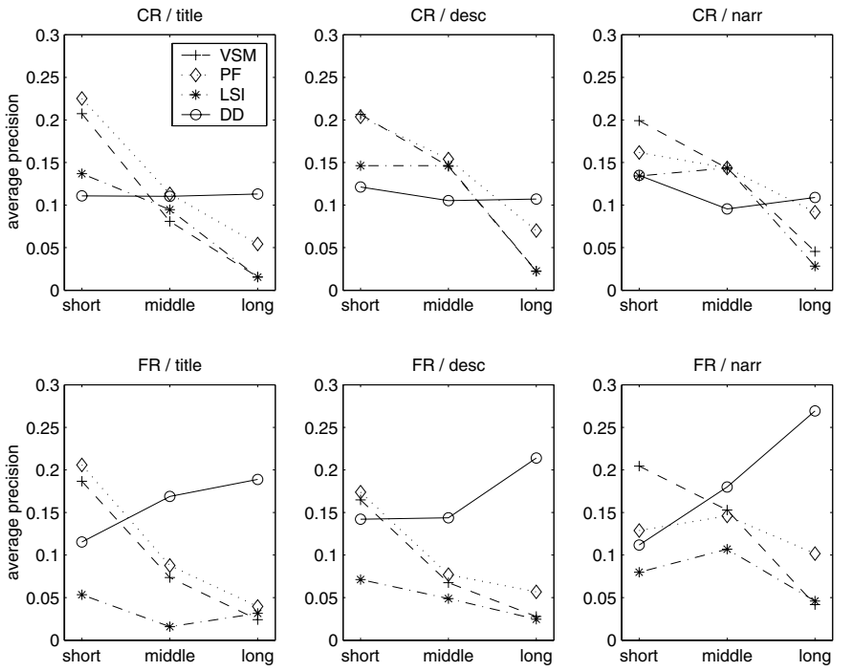


Fig. 4. Average precision for the partitioned collections (horizontal axes : document length).

Table 8. Results of the macro t-test for the partitioned collections.

	methods		CR			FR		
	A	B	title	desc	narr	title	desc	narr
short	DD	VSM	$<$	$\sim$	$<$	$<$	$\sim$	$\ll$
	DD	PF	$\ll$	$<$	$\sim$	$<$	$\sim$	$\sim$
	DD	LSI	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
middle	DD	VSM	$\sim$	$<$	$\ll$	$>$	$>$	$\sim$
	DD	PF	$\sim$	$\ll$	$<$	$>$	$\sim$	$\sim$
	DD	LSI	$\sim$	$\sim$	$\sim$	$\gg$	$\gg$	$\sim$
long	DD	VSM	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	DD	PF	$\gg$	$>$	$\sim$	$\gg$	$\gg$	$\gg$
	DD	LSI	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$

CR collection and "middle" FR collection, the advantage of DD shrank as the query length became longer. These tendencies can also be found in Table 8.

For the "long" FR collection, the difference of the average precision between DD and the best among the other methods was about the same for all query lengths. However, there were disparities in their P-values: the P-value obtained

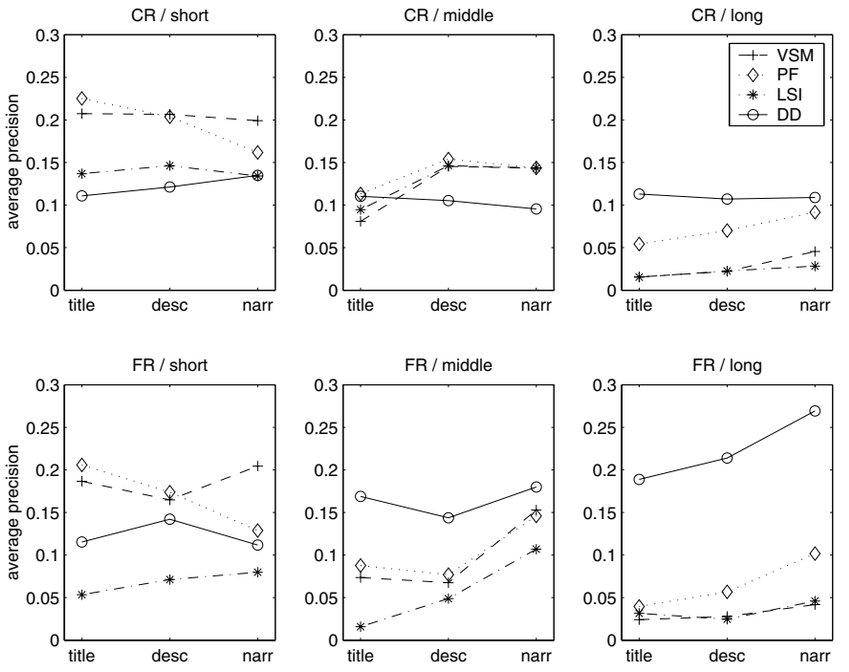


Fig. 5. Average precision for the partitioned collections (horizontal axes : query length).

with the shortest queries (title) was about 10 and 100 times smaller than those with the middle length (desc) and the longest queries (narr), respectively.

From the results obtained from the partitioned collections, we conclude that passage-based document retrieval outperforms conventional methods if relatively lengthy documents are retrieved with short queries. An explanation for this feature of passage-based document retrieval could be as follows. If lengthy documents are retrieved with short queries, it becomes more essential to take into account the *proximity* of query terms, as done only by the passage-based method. In other words, the passage-based method is capable of distinguishing a few query terms which are in the same context (located close to each other in a document) from those occurring in different contexts (far away from each other).

5 Conclusion

We have experimentally evaluated the effect of the length of documents and queries for document retrieval methods. The passage-based method which is capable of ranking documents based on segmented passages has been compared with three conventional document retrieval methods. The results for a variety of document collections show that the passage-based method is superior to conventional methods for longer documents with shorter queries. This feature of

passage-based retrieval is essential if we consider document retrieval as a tool for text mining based on a user's query, since (1) users tend to issue short queries, and (2) available documents are often longer than abstracts.

In order to use passage-based document retrieval as a tool, however, the following things should be further considered. First, the window size appropriate for analyzing documents should be determined automatically. Second, it is required for passage-based document retrieval to work for short documents equivalently to the best conventional method. These issues will be a subject of our future research.

Acknowledgment

This work was supported by the German Ministry for Education and Research, bmb+f (Grant: 01 IN 902 B8).

References

1. M.A.Hearst, Untangling Text Data Mining, in *Proceedings of ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics*, 1999.
2. M.Grobelnik, D.Mladenic and N.Milic-Frayling, Text Mining as Integration of Several Related Research Areas: Report on KDD'2000 Workshop on Text Mining, `http://www.cs.cmu.edu/ dunja/WshKDD2000.html`.
3. J.P.Callan, Passage-level evidence in document retrieval, in *Proc. SIGIR '94*, pp.302-310,1994.
4. G.Salton, A.Singhal and M.Mitra, Automatic text decomposition using text segments and text themes, in *Proc. Hypertext '96*, pp.53-65, 1996.
5. O.de Kretser and A.Moffat, Effective Document Presentation with a Locality-Based Similarity Heuristic, in *Proc. SIGIR '99*, pp.113–120, 1999.
6. K.Kise, H.Mizuno, M.Yamaguchi and K.Matsumoto, On the Use of Density Distribution of Keywords for Automated Generation of Hypertext Links from Arbitrary Parts of Documents, in *Proc. ICDAR'99*, pp.301–304, 1999.
7. R. Baeza-Yates and B.Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley Pub. Co., 1999.
8. C.D.Manning and H.Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.
9. S.Kurohashi, N.Shiraki, and M.Nagao, A Method for Detecting Important Descriptions of a Word Based on Its Density Distribution in Text, *Trans. Information Processing Society of Japan*, Vol.38, No.4, pp.845–853, 1997 [In Japanese].
10. D.Hull, Using Statistical Testing in the Evaluation of Retrieval Experiments, in *Proc. SIGIR '93*, pp.329–338, 1993.
11. Y.Yang and X.Liu, A Re-Examination of Text Categorization Methods, in *Proc. SIGIR '99*, pp.42–49, 1999.
12. `ftp://ftp.cs.cornell.edu/pub/smart/`
13. `http://trec.nist.gov/`

Automated Formulation of Reactions and Pathways in Nuclear Astrophysics: New Results

Sakir Kocabas

Space Engineering Department, ITU, 80626 Maslak, Istanbul, TURKEY
uckoca@itu.edu.tr

Abstract. In this paper we describe some new results from ASTRA, a computational research aid for the formulation and analysis of process explanations in nuclear astrophysics. The program generates fusion and decay reactions for chemical elements by using its knowledge of quantum theory, and from these reactions constructs all theoretically possible reaction chains as process explanations for the nucleosynthesis of heavier elements. Earlier applications of ASTRA generated reactions of the elements and isotopes from hydrogen to oxygen, and found novel reactions and reaction chains for these elements. We have recently extended the system's knowledge base for the elements from oxygen to sulphur. The new applications of ASTRA generated a series of hydrogen burning and helium burning reactions involving heavier elements such as fluorine, neon, sodium, magnesium, aluminium, silicon and sulphur. The program also generated a complete series of carbon, nitrogen and oxygen burning reactions. The new results of ASTRA lead to interesting details about the origin of the elements between oxygen and sulphur.

1 Introduction

As a specialized field of research in artificial intelligence and cognitive science, the computational study of scientific discovery has made important advances in its short history. Early research in the computational study of science was mainly concerned with modeling discoveries from the history of physics, chemistry and biology. The types of discoveries also ranged widely, including numeric laws (e.g. Langley, 1981; Langley, Simon, Bradshaw and Zytkow, 1987), qualitative relations (e.g. Jones 1986), structural models (e.g. Zytkow & Simon, 1986), and process models (e.g. Kulkarni & Simon, 1990). Although important in understanding the conditions of the discoveries, these models produced results already kown to the developers.

In recent years interest increased towards the computational discovery of new scientific knowledge by means of new models (see, Langley, 1998). Among the recent areas of application is the computational design and construction of chemical and nuclear reaction processes. Three examples of such efforts are Hendrickson's (1995) SYNGEN which designs the synthesis of some organic compounds from initial and intermediate compounds, Valdes-Perez's (1995) MECHEM which has found new reaction pathways in catalytic chemistry, and Kocabas and Langley's (1998; 2000) ASTRA system which has found new reactions and pathways in nuclear astrophysics.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 170-181, 2001.

There are other work such as described by Lee, Buchanan, Mattison, Klopman, and Rosenkranz (1995) reporting novel results on whether chemicals cause cancer, and by Mitchell, Sleeman, Duffy, Ingram and Young (1997) with their system DAVICCAND that has found a new numeric relation in metallurgy.

This paper focuses on some of the new results of ASTRA, which has been designed to support scientists in explaining fusion processes, the nucleosynthesis of elements and their relative abundance in stars. The program is a successor of BR-4 (Kocabas & Langley, 1995) which was developed as an integrated model for studying the role of predictions in particle physics, which in turn, was a successor of BR-3 (Kocabas, 1991).

In previous runs, ASTRA was given information about elements and isotopes from hydrogen to oxygen, and the program had generated the reactions and reaction networks for these isotopes. The formation of elements in this range has been extensively studied in nuclear astrophysics. Despite this, the program generated several new reactions and processes of interest to astrophysicists. Recently, we decided to extend the scope of the program to include elements and isotopes from oxygen to sulphur, to see if the program will produce interesting results for the elements in this range. The focus here will be on the new results of ASTRA with an emphasis on the system's abilities as a research tool in astrophysics, rather than its behavior which was described in detail elsewhere (see, Kocabas & Langley, 1998; 2000).

In the next section, we summarize the research topics and methods in nuclear astrophysics, the area of application of ASTRA. Section 3 describes ASTRA in terms of its inputs, outputs, constraints and operations. Section 4 describes the new experimental results of ASTRA, Section 5 discusses these results, and Section 6 discusses related reseach. The paper ends with a summary of the conclusions.

2 The Domain of Nuclear Astrophysics

Nuclear astrophysics is a branch of astrophysics that mainly concerns with the formation of heavier elements from hydrogen (H) and helium (^4He), through a series of fusion and decay processes in stars. Another important concern is the irregularity in the relative abundances of elements, in particular the abundance carbon (^{12}C), nitrogen (^{14}N) and oxygen (^{16}O) compared to lighter elements like lithium (^7Li), beryllium (^9Be) and boron (^{11}B). Exploration of the processes in which the heavier elements from oxygen (^{16}O) to iron (^{56}Fe) are formed is yet another main topic in this field.

According to the current astrophysical theories, stars go through several stages in their lifetimes. The first stage involves 'hydrogen burning' in which hydrogen is transformed into helium. Astrophysicists propose several different pathways (Audouze & Vauclair, 1980, p. 52; Williams, 1991, p. 351) to account for hydrogen burning in stars. Later stages involve more complex reactions and processes such as helium burning, and carbon, nitrogen and oxygen burning.

Astrophysicists explain nucleosyntheses, by first selecting a stellar model in thermal equilibrium which makes certain assumptions about the mass, temperature, density, and the element distribution in the stellar plasma. Then they formulate the

possible and most likely reactions by using several quantum constraints and rate calculations. They then use the reactions with high rates to construct sets of reaction pathways which they call 'reaction networks'.

In our previous work (Kocabas & Langley, 1998; 2000) we examined the results of ASTRA on several research topics concerning the formation of the lighter elements from hydrogen to oxygen. These were: 1) hydrogen burning processes, 2) helium burning processes, 3) formation of carbon, nitrogen and oxygen through hydrogen and helium burning, and other fusion chains, 4) the role of neutrons in such processes, and 5) the anomaly in the relative abundance of the light elements.

In evaluating the results of ASTRA we examined a number of books and journal papers on nuclear astrophysics, notably the following work: Audouze & Vauclair (1980); Clayton (1983); Fowler (1986); Fowler, et al., 1967; Fowler et al., 1975; Harris & Fowler, et al., 1983; Cujec & Fowler, 1980; Kippenhahn & Weigert (1994); Lang (1974); Williams (1991); and Adelberger, E.G., et al. (1998).

3 System Description of ASTRA

Before we describe our application of ASTRA to nuclear astrophysics with some of the earlier and the new results, we briefly describe its inputs, outputs and procedures. A more detailed description can be found in Kocabas and Langley (1998). The program operates in two stages: the first generates all theoretically valid reactions, and the second produces reaction chains as process explanations for the nucleosynthesis of elements.

3.1 Generating Reactions

The first stage of ASTRA takes as input descriptions for a set of elements and isotopes. The current version includes information about 68 such entities. Each entity is characterized in terms of five quantum properties: rest mass (in MeV/c^2), electric charge, spin counts, lepton counts, and baryon counts. ASTRA also has the related rules concerning the conservation of these quantum properties in the reactions.

Using this information, ASTRA generates all collision and decay reactions among these elements that obey the conservation laws, together with their energy emissions, or Q-values, in terms of mega electron volts (MeV). The reactions generated by the program are in the form: $\mathbf{R}_m$ -> $\mathbf{P}_n$, $m = 1,2,3$; $n = 1,2,3$ where $\mathbf{R}_m$ and $\mathbf{P}_n$ are the sets of the reacting and resulting elements respectively, and m and n are the number of elements in the sets. (For $m = 1$, $m=2$ and $m=3$ the formula represents decays, and double and triple collision reactions respectively). An example of the output of this module for the fusion reactions of hydrogen (*H)* with sodium *(^{23}Na)* is as follows:[1]

$$
\begin{array}{lll}
H + {}^{23}Na \text{ --> } {}^{24}Mg & + & 11.68 \text{ (MeV)} \\
H + {}^{23}Na \text{ --> } {}^{24}Na + nu & + & 6.18 \\
H + {}^{23}Na \text{ --> } {}^{20}Ne + {}^{4}He & + & 2.38 .
\end{array}
$$

[1] The reaction formulations of ASTRA are based on neutral atoms. For this reason, there appear minor differences with textbook notations, such as in the second reaction above whose textbook version is $H + {}^{23}Na \text{ --> } {}^{24}Na + /e + nu$, instead of $H + {}^{23}Na \text{ -> } {}^{24}Na + nu$.

In each example, hydrogen and sodium (on the left hand side) combine to form one or more new substances (on the right hand side), along with the total energy emissions in MeV.

For the runs described in this paper, we provided ASTRA with information about the elements from hydrogen to sulphur, their isotopes and a few elementary particles like the electron, proton, neutron and the neutrino with their antiparticles, giving a total of 68 distinct entities. From these, the system generated more than 600 different reactions. We manually eliminated minor variations such as *[3]He + [9]Be --> [12]C + e + /e* and *[3]He + [9]Be --> [12]C + nu + /nu*, leaving 472 reactions that included 344 fusion reactions and 28 decays.

3.2 Generating Reaction Chains

Taking as input the reactions generated by the first stage, ASTRA generates the reaction chains for an element *E* from a small set of basic elements/isotopes (**E**) that we assume as given. The system uses a depth-first, backward chaining search to construct the reaction chains. On the first step, ASTRA finds those reactions that give as an output the final element *E*. Upon selecting one of these reactions, *R*, it recursively finds those reactions that give as an output one of more *R*'s input elements. The algorithm continues this process, halting its recursion when it finds a reaction chain for which all the reacting elements are in (**E**), or when it cannot find a reaction off which to chain. ASTRA generates all possible reaction chains in this systematic manner.

4 New Results of ASTRA

In this section we report the new results of our tests with ASTRA concerning hydrogen-, helium-, carbon- and oxygen-burning reactions. We start with proton, electron and neutron capture reactions of heavier elements such as oxygen, fluor, neon, sodium, magnesium, aluminium, silicon and phosphorus.

4.1 Proton, Electron, and Neutron Captures

Proton captures are an important class of exothermic reactions that also take part in processes transforming hydrogen into helium as will be described below. Proton capture by an atomic nucleus turns it into another element with one higher atomic number. ASTRA finds 33 examples of proton captures given in astrophysics literature (e.g., Fowler, et al., 1967, 1975, 1983) for elements from hydrogen to oxygen (^{16}O), and 20 more for elements from oxygen to sulphur.

ASTRA's first stage predicts that all elements from hydrogen to sulphur (^{32}S), with the exception of ^{4}He, participate in exothermic proton capture. The program produces 46 such reactions for elements from hydrogen to oxygen, including all 33 examples we have found in texts, but also 13 others which we have not seen in astrophysics texts that we examined. The program also finds 72 proton captures for elements from oxygen (^{16}O) to sulphur (^{32}S), including the 20 such reactions cited in the same literature. Three examples of such proton captures are,

$$H + {}^{19}F \text{ --> } {}^{20}Ne$$
$$H + {}^{23}Na \text{ --> } {}^{24}Mg$$
$$H + {}^{27}Al \text{ --> } {}^{28}Si.$$

In these reactions, proton captures by fluorine, sodium and aluminium, transforms them into neon, magnesium and silicon, respectively.

Also, all the isotopes from oxygen to sulphur, with the exception of the isotopes of neon and magnesium, participate in exothermic proton captures that produce helium (^4He). Three examples to such reactions are,

$$H + {}^{19}F \text{ --> } {}^{4}He + {}^{16}O$$
$$H + {}^{23}Na \text{ --> } {}^{4}He + {}^{20}Ne.$$
$$H + {}^{27}Al \text{ --> } {}^{4}He + {}^{24}Mg.$$

Electron capture reactions are weak interactions in which an electron is absorbed by the atomic nucleus to be transformed into one with a smaller atomic number. In the process, the electron is combined with a proton in the nucleus, effectively transforming it into a neutron with the emission of a neutrino:

$$e + p \text{ --> } n + nu.$$

ASTRA's first stage produces 6 electron capture reactions for elements from hydrogen to oxygen of which only the one just given appears in astrophysics texts. The program also found 8 electron capture reactions for elements from oxygen to sulphur, none of which we have seen in the texts.

In neutron capture, an element combines with a neutron to form a heavier isotope of the same element. We found 17 neutron captures for elements from hydrogen to oxygen in the literature, while ASTRA predicts 59 such reactions that are theoretically possible for the same elements. Some examples of these reactions can be found in Kocabas and Langley (1998). Recent runs of the system generated 76 reactions for elements from oxygen to sulphur. Three examples of such neutron capture reactions are,

$$n + {}^{18}F \text{ --> } {}^{19}F$$
$$n + {}^{22}Na \text{ --> } {}^{23}Na$$
$$n + {}^{31}S \text{ --> } {}^{32}S.$$

Here, as indicated above, in each case the nucleus that absorbs the neutron turns into a heavier isotope of the same element.

4.2 Hyrogen Burning Processes

The transformation of hydrogen into helium in a series of nuclear processes which take place in main sequence stars i the principal source of energy. The standard reaction chains given in astrophysics texts (e.g. Audouze & Vauclair, 1980, p. 52; Williams, 1991, p. 351) for helium synthesis in such stars are the hydrogen-burning processes called "proton-proton" or *pp* chains. Other hydrogen burning reactions that

appear in texts involve heavier elements carbon, nitrogen and oxygen, and the pathway is called the CNO-chain. ASTRA produces all known CNO-chains, in addition to one viable variant using the electron capture of ^{13}N (see, Kocabas & Langley, 1998).

We have tested ASTRA on hydrogen burning reactions involving the elements heavier than oxygen. Such reactions are hypothesized to occur in stars several times larger than the sun. The program found four hydrogen burning chains involving the elements fluorine, neon, sodium, magnesium, silicon, phosphorus and sulphur. One of these processes is

$$
\begin{aligned}
&H + {}^{24}Mg \rightarrow {}^{25}Mg + nu\\
&H + {}^{25}Mg \rightarrow {}^{26}Al\\
&H + {}^{26}Al \rightarrow {}^{27}Si\\
&{}^{27}Si + e \rightarrow {}^{27}Al + e + nu\\
&H + {}^{27}Al \rightarrow {}^{24}Mg + {}^{4}He\\
&\text{----------}\\
&4\,H \rightarrow {}^{4}He + 2\,nu\ .
\end{aligned}
$$

In this process four hydrogen atoms in effect, transform into one helium atom, while two neutrinos are also emitted. We did not see any of these processes in the texts that we examined, but we presume that they are known to astrophysicists.

4.3 Helium Burning Processes

The origin and the relative abundance of carbon and oxygen has been one of the main concerns of astrophysics. The standard account (e.g., Fowler, 1986, pp. 5-6) relies on the process of helium-burning, in which helium nuclei react to form carbon and oxygen in the following steps:

$$
\begin{aligned}
&{}^{4}He + {}^{4}He \rightarrow {}^{8}Be\\
&{}^{4}He + {}^{8}Be \rightarrow {}^{12}C\\
&{}^{4}He + {}^{12}C \rightarrow {}^{16}O\ .
\end{aligned}
$$

In its earlier runs, ASTRA found an alternative to this process which astrophysicists qualified as more likely in neutron-rich stellar media (see, Kocabas & Langley, 2000).

ASTRA finds 25 exothermic helium burning reactions involving the range of elements from oxygen to silicon, including the 16 such reactions cited in the texts. Some of these reacitons are,

${}^{4}He + {}^{16}O \rightarrow {}^{20}Ne$	*5.16* (MeV)
${}^{4}He + {}^{19}F \rightarrow {}^{23}Na$	*10.5*
${}^{4}He + {}^{19}F \rightarrow H + {}^{22}Ne$	*1.72*
${}^{4}He + {}^{20}Ne \rightarrow {}^{24}Mg$	*9.3*
${}^{4}He + {}^{22}Ne \rightarrow {}^{26}Mg.$	*10.6*

$^{4}He + {}^{23}Na \rightarrow {}^{27}Al$	10.2
$^{4}He + {}^{23}Na \rightarrow /nu + {}^{27}Si$	5.4
$^{4}He + {}^{23}Na \rightarrow H + {}^{26}Mg$	1.82
$^{4}He + {}^{24}Mg \rightarrow {}^{28}Si$	10.1
$^{4}He + {}^{25}Mg \rightarrow {}^{29}Si$	11.2
$^{4}He + {}^{25}Mg \rightarrow nu + {}^{29}Al$	7.5
$^{4}He + {}^{25}Mg \rightarrow n + {}^{28}Si$	2.73
$^{4}He + {}^{26}Mg \rightarrow {}^{30}Si$	10.8
$^{4}He + {}^{26}Mg \rightarrow /nu + {}^{30}P$	6.5
$^{4}He + {}^{26}Mg \rightarrow n + {}^{29}Si$	0.13
$^{4}He + {}^{27}Al \rightarrow {}^{31}P$	9.7
$^{4}He + {}^{27}Al \rightarrow /nu + {}^{31}S$	4.2
$^{4}He + {}^{27}Al \rightarrow H + {}^{30}Si$	2.42
$^{4}He + {}^{28}Si \rightarrow {}^{32}S$	6.9
$^{4}He + {}^{28}Si \rightarrow nu + {}^{32}P$	4.6
$^{4}He + {}^{29}Si \rightarrow {}^{33}S$	7.2

Among these reactions those that emit neutrinos (*nu* and */nu*) are weak interactions which are much slower than the other alpha capture reactions. Astrophysicists generally ignore the weak reactions for their slow rates, except in processes that rely on such weak reactions.

A careful comparison of the proton capture, neutron capture and helium burning reactions produced by ASTRA with the natural abundances of the elements from oxygen to sulphur in the CRC Handbook (80th ed., D.R.Lide, 1999-2000) reveals an interesting result: The elements fluorine, neon, sodium, magnesium, silicon, phosphorus and sulphur in the solar system must have been formed by alpha capture processes, rather than proton or neutron captures. This is because, the stable isotope abundances of these elements indicate a parallelism with the stepwise alpha-capture (helium burning) of the stable lighter isotopes of the elements in the series (see Table 1). Indeed, the two alpha capture chains (^{16}O, ^{20}Ne, ^{24}Mg, ^{28}Si, ^{32}S and ^{19}F, ^{23}Na, ^{27}Al, ^{31}P) contain the most abundant isotopes of these elements. These processes may have been accompanied by carbon, nitrogen and oxygen burning processes which produce ^{24}Mg, ^{28}S and ^{32}S respectively as shown in the next subsection.

Although proton capture reactions explain the relative abundance of ^{19}F, ^{20}Ne, ^{23}Na, ^{24}Mg, ^{27}Al, ^{28}Si, and ^{32}S, they fail to explain the relative abundance of ^{31}P. Similarly, neutron capture reactions fail to explain the relative abundances of ^{20}Ne, ^{24}Mg and ^{28}Si. Yet, stepwise alpha capture explains the relative abundances of all the isotopes in the series. We are currently investigating the astrophysical literature on the origins of the elements from fluorine to sulphur before claiming any novelty on this issue.

Table 1. Relative abundances of some isotopes for elements from oxygen to sulphur.

isotope	% abundance	isotope	% abundance
^{16}O	99.76	^{18}O	0.2
^{19}F	100	^{18}F	0
^{20}Ne	90.48	^{22}Ne	9.25
^{23}Na	100	^{22}Na	0
^{24}Mg	78.99	^{26}Mg	11.01
^{27}Al	100	^{26}Al	0
^{28}Si	92.23	^{29}Si	4.67
^{31}P	100	^{30}P	0
^{32}S	95.0	^{34}S	4.21

4.4 Carbon, Nitrogen, and Oxygen Burning

Carbon burning, in which two carbon atoms fuse together to produce heavier elements, takes place after the helium burning stage in a star. ASTRA finds four carbon burning reactions which produce the elements neon, sodium, and magnesium:

$^{12}C + {}^{12}C \rightarrow {}^{24}Mg + 14.4$ (MeV)
$^{12}C + {}^{12}C \rightarrow nu + {}^{24}Na + 8.9$
$^{12}C + {}^{12}C \rightarrow H + {}^{23}Na + 2.72$
$^{12}C + {}^{12}C \rightarrow {}^{4}He + {}^{20}Ne + 5.1$

In nitrogen burning, two nitrogen atoms fuse together to form elements ranging from oxygen to silicon. ASTRA finds 10 such reactions:

$^{14}N + {}^{14}N \rightarrow {}^{28}Si + 27.82$ (MeV)
$^{14}N + {}^{14}N \rightarrow nu + {}^{28}Al + 23.12$
$^{14}N + {}^{14}N \rightarrow n + {}^{27}Si + 10.65$
$^{14}N + {}^{14}N \rightarrow H + {}^{27}Al + 16.24$
$^{14}N + {}^{14}N \rightarrow D + {}^{26}Al + 5.52$
$^{14}N + {}^{14}N \rightarrow {}^{3}He + {}^{25}Mg + 4.52$
$^{14}N + {}^{14}N \rightarrow {}^{4}He + {}^{24}Mg + 17.72$
$^{14}N + {}^{14}N \rightarrow {}^{8}Be + {}^{20}Ne + 8.32$
$^{14}N + {}^{14}N \rightarrow {}^{12}C + {}^{16}O + 10.46$
$^{14}N + {}^{14}N \rightarrow {}^{13}N + {}^{15}N + 0.72$

Finally, ASTRA formulates the following oxygen burning reactions in which two oxygen atoms fuse together in exothermic reactions, and the elements magnesium, silicon, phosphorus and sulphur are generated:

$^{16}O + {}^{16}O \rightarrow {}^{32}S$	+ 17.12	(MeV)
$^{16}O + {}^{16}O \rightarrow nu + {}^{32}P$	+ 14.82	
$^{16}O + {}^{16}O \rightarrow n + {}^{31}S$	+ 2.05	
$^{16}O + {}^{16}O \rightarrow H + {}^{31}P$	+ 8.34	
$^{16}O + {}^{16}O \rightarrow {}^{4}He + {}^{28}Si$	+ 10.22	
$^{16}O + {}^{16}O \rightarrow {}^{8}Be + {}^{24}Mg$	+ 0.02	

Carbon, nitrogen and oxygen burning reactions happen only in massive stars as they require higher energies to initiate. The astrophysics texts that we examined mention only a few of these reactions, such as $^{12}C + {}^{12}C \rightarrow {}^{24}Mg$, $^{14}N + {}^{14}N \rightarrow {}^{28}Si$, and $^{16}O + {}^{16}O \rightarrow {}^{32}S$, while ASTRA provides a full account of such reactions.

5 Discussion of Results

We have compared ASTRA's earlier outputs involving the elements from hydrogen to oxygen to those available in astrophysics texts (Clayton, 1983; Audouze & Vauclair, 1980; Kippenhahn & Weigert, 1994; Fowler et al., 1967, 1975, 1983; Cujec & Fowler, 1980; Adelberger, E.G., et al. (1998), and discussed some of its results with astrophysicists. We received encouraging comments from domain experts on the earlier outputs (see, Kocabas & Langley, 2000). However, the reactions and processes of the light elements have already been studied extensively by nuclear astrophysicists. For this reason, we decided to extend the scope of the program to investigate the reactions and the processes of the elements from oxygen to sulphur.

The ASTRA program can handle a very large volume of data for constructing reactions and reaction networks. Astrophysicists normally formulate the reactions by hand, and construct the reaction networks by focusing on the more likely reactions by using certain domain criteria. It is in this way the hydrogen and helium burning processes involving the lighter elements have been dealt with extensively in the current literature. But as the number of possible reactions increase rapidly for the heavier elements, a complete analysis of the reactions and processes can only be carried out with the aid of a computational tool such as our program. Although we tested ASTRA on the reactions of the elements from hydrogen (*H)* to sulphur (*^{32}S)* with some interesting results, we plan to extend the system for exploring the reactions of heavier elements from sulphur to iron (^{56}Fe) and further, which take place in stellar and interstellar processes.

The understanding of the nuclear processes in which the chemical elements are formed is important in more ways than one, as this provides detailed information about the stellar and interstellar conditions that produced these elements. This is why cosmologists and astronomers are also very much interested in these processes as well as nuclear astrophysicists. We have described in Section 4, how an analysis of the reactions and the reaction processes produced by ASTRA and the natural abundances of chemical elements and isotopes, can lead to a detailed picture of the conditions in which these elements are formed.

Astrophysicists use reaction rates to rule out slower reactions from their reaction networks. The current version of ASTRA can use reaction rates to rule out candidates, retaining only those reactions with the highest rates to construct reaction networks. But the rate for each reaction must be given by the user, the program cannot calculate them. We attempted to incorporate the rate calculation in ASTRA recently but decided

not to gon on with this, because of the complexities involved. Rate calculations are based on the reaction cross-sections and element concentrations in stellar media. Astrophysicists first construct a model of the star by making a number of assumptions about the star size, mass, temperature, pressure and element distribuiton. Stellar plasma are also treated in several layers through which element compositions, dominant reactions and processes change.

Although ASTRA can search a much larger space of reactions and processes than can human scientists. We did not meet any problems with it for the elements and isotopes from hydrogen to sulphur involving 68 distinct entities. We have yet to see if we will need to constrain the scope of the reactions for the elements from hydrogen to iron. We plan to extend the program to investigate the reactions of the elements from sulphur to iron. Meanwhile we will continue to investigate the literature about the origins of heavier elements in the solar system.

6 Related Research

The ASTRA system has evolved from our previous work in computational study of discoveries in particle physics with BR-4 (Kocabas & Langley, 1995), which models the discoveries in this field by prediction and theory revision. BR-4 inherits some of its capabilities from its predecessor BR-3 (Kocabas, 1991), which in turn descends from STAHL (Zytkow & Simon, 1986), and STAHLp (Rose & Langley, 1986) which modelled qualitative discovery in chemistry.

Our system shares goals and techniques with more recent systems MECHEM (Valdes-Perez, 1995) designed to discover new reaction mechanisms in catalytic chemistry, and SYNGEN (Hendrickson, 1995) which constructs pathways for the synthesis of complex organic chemicals from simpler constituents. There are many similarities between ASTRA and MECHEM in terms of the tasks they perform. Both systems produce reactions and reaction mechanisms in large search spaces, and both are designed as computational aids for scientists. But the two systems differ in their inputs and outputs. MECHEM receives as input the initial and final chemical substances and generates all the simple reaction pathways using a set of constraints on chemical reactivity. Similarly, ASTRA uses a set of quantum constraints to formulate the reactions from which it constructs the reaction links for each element until the final element is reached. The reaction links in a chain constitute what is called by astrophysicists 'the reaction network'.

ASTRA has to deal with a large number of entities (elementary particles, elements and their isotopes), and even much larger number of reactions of these entities, to construct valid reactions and reaction chains, while MECHEM has a relatively smaller search space in its domain of application. MECHEM's reaction pathways are lists of reaction steps normally with at most two reactants and two products. In contrast, the reactions of ASTRA can have from one to three entities in both sides.

As to the comparison between our system and SYNGEN, the latter addresses the synthesis of organic chemicals, where one needs to determine reaction paths and the initial substances, through a set of known intermediate substances. The constraints of SYNGEN are more similar to those used by MECHEM though they operate in different fields of chemistry. Our program differs from these systems in its field of application and the types of constraints used.

7 Conclusions

In this paper we described the new results of ASTRA, a computational tool which formulates reactions and reaction chains for researchers in nuclear astrophysics. The system determines all valid reactions for a given set of elements, isotopes and particles using a set of quantum constraints. The system also generates all reaction pathways for an element starting from a set of lighter elements. ASTRA generates all reactions we have seen in the astrophysics literature involving proton, electron and neutron captures, and helium, carbon, nitrogen and oxygen burning. ASTRA also reproduces all reaction chains that scientists have proposed for the synthesis of helium, carbon, nitrogen and oxygen in stellar media. But many of the valid reactions and reaction chains that the system generates do not appear in the related scientific literature. The domain experts that we have contacted suggested that some of these results carry theoretical interest for certain stellar models, but the vast majority of the reaction chains would be ignored by astrophysicists for their low rates.

Earlier we decided to incorporate the rate calculations in the ASTRA system, but later abandoned this project because of the complexities involved. Instead, we focused on extending the system's knowledge base to investigate the reactions and processes of the heavier elements. Given information about 32 more elements and isotopes from oxygen to sulphur, amounting to a total of 68 distinct entities, the program generated all the proton, electron, neutron capture reactions and all the helium, carbon, nitrogen burning reactions. A close comparison of these reactions with the stability and natural abundances of the 32 istopes between oxygen and sulphur indicated that the stable isotopes in this range must have been formed by exothermic alpha capture reactions accompanied by carbon, nitrogen and oxygen burning rather than proton or neutron capture reactions. We are currently investigating the literature for any scientific record on this issue.

References

Adelberger, E.G., et al. (1998). Solar fusion cross sections. *Reviews of Modern Physics*, vol. 70, No. 4. Pp 1266-1291.

Audouze, J., & Vauclair, S. (1980). *An introduction to nuclear astrophysics*. Holland: D. Riedel.

Clayton, D.D. (1983). *Principles of Stellar Evolution and Nucleosynthesis*. Chicago: The University of Chicago Press.

Cujec, B. & Fowler, W.A. (1980). Neglect of D, T, and 3He in advanced stellar evolution. *The Astropysical Journal*, 236: 658-660.

Feigenbaum, E. A., Buchanan, B.G., Lederberg, J. (1971). On generality and problem solving: A case study using the DENDRAL program. In *Machine Intelligence* (vol. 6). Edinburgh: Edinburgh University Press.

Fowler, W.A. (1986). The synthesis of the chemical elements carbon and oxygen. In S.L. Shapiro & S.A. Teukolsky (Eds.), *Highlights of modern astrophysics*. New York: John Wiley & Sons.

Fowler, W.A., Caughlan, G.R., and Zimmermann, B.A. (1967). Thermonuclear Reaction Rates. *Ann. Rev. Astron. Astrphysics*, **5**, 525-570.

Fowler, W.A., Caughlan, G.R., and Zimmermann, B.A. (1975). Thermonuclear Reaction Rates. *Ann. Rev. Astron. Astrphysics*, **13**, 69-112.

Harris, M.J., Fowler, W.A. Caughlan, G.R., and Zimmermann, B. (1983). Thermonuclear reaction rates. *Ann. Rev. Astron. Astrophysics*, **21**, 165-176.

Hendrickson, J.B. (1995). Systematic synthesis design: The SYNGEN program. *Working Notes of the AAAI Spring Symposium on Systematic Methods of Scientific Discovery* (pp. 13-17). Stanford, CA: AAAI Press.

Jones, R. (1986). Generating predictions to aid the scientific discovery process. *Proceedings of the Fifth National Conference on Artificial Intelligence,* pp. 513-517, Philadelphia: Morgan Kaufmann.

Kippenhahn, R. and Weigert, A. (1994). *Stellar Structure and Evolution*. London: Springer-Verlag.

Kocabas, S. (1991). Conflict resolution as discovery in particle physics. *Machine Learning*, **6**, 277-309.

Kocabas, S., & Langley, P. (1995). Integration of research tasks for modeling discoveries in particle physics. *Working notes of the AAAI Spring Symposium on Systematic Methods of Scientific Discovery* (pp. 87-92). Stanford, CA: AAAI Press.

Kocabas, S. & Langley, P. (1998). Generating process explanations in nuclear astrophysics. *Proceedings of the ECAI-98 Workshop on Machine Discovery* (pp. 4 -9), Brighton, UK.

Kocabas, S. & Langley, P. (2000). Computer generation of process explanations in nuclear astrophysics. *International Journal of Human-Computer Studies*, **53**, 1149-1164, Academic Press.

Kulkarni, D., & Simon, H.A. (1990). Experimentation in machine discovery. In J. Shrager & P. Langley (Eds.), *Computational models of scientific discovery and theory formation*. San Mateo, CA: Morgan Kaufmann.

Lang, K.R. (1974*). Astrophysical formulae: A compendium for physicists and astrophysicists*. New York: Springer-Verlag.

Langley, P. (1981). Data-driven discovery of physical laws. *Cognitive Science*, **5**, 31-54.

Langley, P. (1998). The computer-aided discovery of scientific knowledge. *Proceedings of* the 1*st International Conference on Discovery Science,* Fukuoka, Japan: Springer.

Langley, P., Simon, H.A., Bradshaw, G.L., & Zytkow, J.M. (1987). *Scientific Discovery: Computational explorations of the creative processes*. Cambridge, MA: MIT Press.

Lee, Y., Buchanan, B.G., Mattison, D.R., Klopman, G., & Rosenkranz, H.S. (1995). Learning rules to predict rodent carcinogenicity. *Machine Learning*, **30**, 217-240.

Lide, D.R. (Ed.). (1999-2000). *CRC handbook of chemistry and physics* (80th ed.). Florida: CRC Press.

Mitchell, F., Sleeman, D., Duffy, J.A., Ingram, M.D., & Young, R.W. (1997). Optical basicity of metallurgical slags: A new computer-based system for data visulisation and analysis. *Ironmaking and Steelmaking,* **24**, 306-320.

Rose, D. & Langley, P. (1986). Chemical discovery as belief revision. *Machine Learning*, 1, 423-451.

Valdes-Perez, R.E. (1995). Machine discovery in chemistry: New results. *Artificial Intelligence,* **74**, 191-201.

Williams, W.S.C. (1991). *Nuclear and Particle Physics*. Oxford: Clarendon Press.

Zytkow, J.M., & Simon, H.A. (1986). A theory of historical discovery: The construction of componential models. *Machine Learning*, **1**, 107-137.

An Integrated Framework for Extended Discovery in Particle Physics

Sakir Kocabas[1] and Pat Langley[2]

[1] Space Engineering Department, ITU
80626 Maslak, Istanbul, Turkey
ukoca@itu.edu.tr

[2] Institute for the Study of Learning and Expertise
2164 Staunton Court, Palo Alto, CA 94306 USA
langley@isle.org

Abstract. In this paper we describe BR-4, a computational model of scientific discovery in particle physics. The system incorporates operators for determining quantum values of known particles, formulating new quantum properties, positing new particles, and predicting reactions among particles. BR-4 carries out heuristic search guided by constraints that its theory be consistent and complete with respect to observed reactions. We show that this control scheme is sufficient to model, with some manual intervention, an extended period in the history of particle physics, including the discovery of the neutrino and the postulation of baryon, lepton, and electron numbers. In closing, we compare BR-4 to other discovery systems and suggest directions for future research.

1 Introduction and Motivation

Computational research on scientific discovery falls into two broad categories. The first, typified by the work of Langley, Simon, Bradshaw, and Żytkow (1987), focuses on modeling the processes responsible for discoveries from the history of science. The second approach, exemplified by the work of Valdés-Pérez (1995) and Mitchell, Sleeman, Duffy, Ingram, and Young (1997), uses computational methods to discover new scientific knowledge. These two approaches share many ideas, and both have made valuable contributions to discovery science, but they have distinct goals and criteria for evaluation.

In this paper we describe results within the first, historical, approach to scientific discovery. Like Nordhausen and Langley (1993), we believe that there has been important progress in this area, but that most previous models have focused on one aspect of the scientific process to the exclusion of others. Like them, our goal has been to extend earlier models to account for a broader range of scientific enquiry during an extended period in science. We have not tried to model the processes in detail or to craft a precise theory of human cognition, but rather to provide an abstract but unified account of major activities and their order of occurrence. This has required us to develop an integrated framework that combines discovery mechanisms in a coherent way.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 182–195, 2001.

Nordhausen and Langley's work addressed empirical discovery in physics and chemistry, which led their IDS system to integrate mechanisms for forming taxonomies, finding qualitative laws, and detecting numeric relations. We have focused instead on the more theory-laden domain of particle physics, so that our BR-4 system integrates processes for constructing and revising structural theories, detecting and formulating problems, generating new theoretical terms, and predicting new events.

In the next section we present our integrated framework for scientific discovery and its implementation in BR-4. After this, we consider four examples from the history of particle physics, showing for each how the system simulates discoveries made during the period. These case studies include the postulation of the neutrino, the prediction of various reactions, the proposal of baryon and lepton numbers, and the discovery of electron and muon numbers. In closing, we review related computational work on discovery and consider directions for extending our framework.

2 A Framework for Discovery in Particle Physics

In this section we present a computational framework for explaining the processes that support scientific discovery in particle physics, starting with an analysis of the task. We then turn to the representational assumptions that underlie our framework, the heuristics that drive the discovery process, and the search algorithm that our model, BR-4, uses to explore the space of theories.

2.1 The Discovery Task

Particle physics studies the nature of elementary particles – the building blocks of matter – and interactions among these entities. The basic phenomena in this field take the form of reactions, similar in many ways to those found in chemistry. For instance, one such 'observed' reaction (typically inferred from tracks in cloud chambers) is $p + p \rightarrow p + n + \pi$, where the symbols p, n, and π represent the proton, neutron, and pion particles, respectively.

As in chemistry, physicists require that reactions among elementary particles obey certain conservation laws. One of the main tasks in particle physics concerns the assignment of values for *quantum properties* such that observed reactions conserve those properties. For example, the above reaction conserves the quantum property of electric charge, provided we assign the accepted charges 1 to p, 0 to n, and 1 to π. Other assignments are possible for this reaction, but they would not work for other particles and their observed interactions.

The notion of conservation also explains why some particle reactions are never observed. For example, proton decay, as in the reaction $p \rightarrow \bar{e} + \gamma$, has never been seen, despite its conservation of electric charge. However, one can explain its absence by positing that it fails to conserve another quantum property, the baryon number. Thus, another central task in particle physics involves explaining missing reactions by postulating new quantum properties.

Other activities include the inference of new particles, either on theoretical or empirical grounds, and the prediction of reactions that involve these particles in ways that satisfy known conservation laws. Testing such predictions leads into the realm of experimental particle physics, which we will not address here. But the above pursuits cover a wide range of the behaviors that occur in this scientific field.

2.2 Discovery Operators and Internal Representation

The above analysis of the discovery task suggests that four basic operators play a central role in particle physics. First, for a given set of particles, quantum numbers, and observed reactions, we must be able to determine a set of quantum values that satisfy conservation for those reactions. Second, we must be able to posit new quantum properties that account for the absence of unobserved reactions. Third, we require an operator that posits new particles and their role in known reactions. Finally, we need some mechanism for predicting reactions that have not yet been observed, but that follow from the current theory. We have incorporated these operators into the BR-4 model, where they support the process of theory formation and revision.

Operators of this sort must alter some internal representation that contains hypotheses about the particles, properties, and reactions that exist, and that also indicates specific quantum values for each pair of property and particle. This representation can take many forms, but, following Valdés-Pérez, Żytkow, and Simon (1993), we can view it as two related matrices. One matrix lists particles against quantum properties, with each matrix entry specifying the value for a specific particle on a specific property. The other matrix lists particles against reactions, with an entry containing the total number of times the particle occurs in the reaction. Our operator for determining quantum values alters entries in the first matrix, whereas the other operators each extend one or both matrices along one of their dimensions. In our examples, we will use the matrix notation to specify the properties of particles but not the reactions in which they occur, since the latter matrix would be largely empty.

2.3 Heuristics for Consistency and Completeness

Naturally, simply formulating the problem in this manner does not solve it. Given P particles and Q quantum properties with V values each, there are $V^{Q \cdot P}$ possible assignments of values to particle-property pairs. For small values of P, Q, and V, one could search this space exhaustively, but recall that one must also consider different numbers for these parameters themselves (i.e., different size matrices). In general, constrained search is preferable to blind search, and we have incorporated a number of heuristics into the BR-4 system that focus its attention in useful directions.

First, the system considers simpler theories first, starting with one that contains only directly 'observable' particles, quantum properties for which there exists separate evidence (such as electric charge), and a few observed reactions.

Second, BR-4 alters this theory only when it encounters evidence of some deficiency, and then it considers only those operators that promise to repair the problem. Finally, the model uses constraints on the problem domain, such as conservation, to limit the search within the space of repairs.

More specifically, BR-4's approach to discovery in particle physics relies on the notions of *consistency* and *completeness* to constrain the reasoning process. For example, the operator for determining quantum values applies only when the system detects that an observed reaction is inconsistent with some conservation law. In this case, it carries out a depth-first search through the space of values, continuing until it encounters a value combination that violates conservation, in which case it backtracks. When this process is complete, the resulting quantum values are guaranteed to be consistent with all reactions observed so far. To keep the process tractable, BR-4 considers only the values 0, 1, and −1 during its search.[1]

In some cases, the above revision process cannot eliminate the inconsistency, either because no combination of property values leads to conservation across all observed reactions or because the quantum values are determined experimentally (as for the spin number). This condition leads BR-4 to revise the unbalanced reaction by adding a 'hidden' particle in either the input or output, positing that it actually takes part in the reaction but for some reason is not directly observable. The system then computes the property values that would balance the reaction and associates them with the new particle.

The incompleteness constraint leads to complementary behavior. When BR-4 finds that its current theory fails to rule out a reaction that does not occur, it introduces a new quantum property that is *not* conserved by this reaction but that is conserved by those it has observed. Determining the values of this property requires search, first for the values of particles in the missing reaction (constrained to satisfy an inequality), and then an embedded search for the values of other particles (constrained to satisfy equalities corresponding to observed reactions). As before, if the system arrives at a partial combination of values that rules out an observed reaction or fails to eliminate the unobserved one, it considers alternative paths until it finds an acceptable set. In both searches, BR-4 considers smaller absolute values before turning to larger ones.

We can extend the notion of incompleteness to include theories that do not explicitly specify all reactions that follow from them, as occurs when one postulates a new particle. In this situation, BR-4 systematically generates all possible reactions of the new particle involving one, two, or three other known particles. Some of these reactions take the form of decays, whereas others involve collisions among particles. For each such tentative reaction R, the system predicts that R will occur if it conserves all known properties, and predicts that the reaction will not occur otherwise.

[1] Physicists assign to the spin property not only integers like 0 and 1, but also values like $\frac{1}{2}$ and $\frac{3}{2}$. BR-4 also considers these values for this property and, like physicists, calculates the spin number using group theory.

Table 1. The quantum values for elementary particles known (a) in 1930, prior to experimental detection of the neutron, and (b) after postulation of the neutrino.

	Particle	mass	charge	spin
(a)	γ	0.00	0	1
	e	0.51	-1	$\frac{1}{2}$
	p	938.26	1	$\frac{1}{2}$
	$\bar{e}$	0.51	1	$\frac{1}{2}$
(b)	n	939.55	0	$\frac{1}{2}$
	ν	0.00	0	$\frac{1}{2}$

3 Illustrative Examples from Particle Physics

In this section we consider four examples of discovery from the history of particle physics, involving the neutrino, baryon and lepton numbers, and electron and muon numbers. In each case, we recount the main historical events, and then examine BR-4's behavior when presented with similar observations. Our historical treatment is based upon a number of sources on particle physics, including Griffiths (1987), Ne'eman and Kirsh (1986), Omnes (1970), Pais (1986), and Trefil (1980).

3.1 Discovery of the Neutrino

Until the early 1930's, scientists had observed only a few elementary particles, shown in Table 1 (a) along with their mass and their values on three conserved quantum properties – energy, charge, and spin. The known reactions were also limited to a small set: $p+p \rightarrow p+p$, $e+\bar{e} \rightarrow \gamma$, and $\gamma \rightarrow e+\bar{e}$. This situation changed after Chadwick's experimental detection of the neutron in 1932, which also clarified another outstanding issue (Giancoli, 1995).

Much earlier, physicists had observed beta decay, a process in which an element emits an electron and is transformed into another element with a higher atomic number. This transformation appeared to violate conservation of both energy and spin, leading Bohr to suggest that these properties are truly not conserved within the nucleus. However, in 1930, Pauli proposed another explanation – that beta decay also emitted another particle that was difficult to detect.

Chadwick's experiments also revealed neutron decay, $n \rightarrow p + e$, which occurs in about 800 seconds on free neutrons. Like beta decay, this reaction appeared to violate energy and spin conservation, but in simplified form. Again, Pauli's account avoided this problem by postulating a new particle, also generated during the decay reaction, that would balance out the missing energy and spin. In 1934, Fermi formalized this proposal for the *neutrino*, which he posited as having zero rest mass, no electrical charge, and a spin of one half.

Table 2. Particle reactions that were (a) observed and (b) not observed in experiments after the introduction of the particles in Table 1 (b).

(a) Observed reactions	(b) Unobserved reactions
$p + p \rightarrow p + p$	$p \rightarrow \bar{e} + \gamma$
$e + \bar{e} \rightarrow \gamma$	$p \rightarrow \bar{e} + e + \bar{e}$
$\gamma \rightarrow e + \bar{e}$	$p \rightarrow \bar{e} + \gamma + \gamma$
$\gamma + p \rightarrow e + \bar{e} + p$	
$n \rightarrow p + e + \nu$	

Given the four reactions above and the quantum numbers in Table 1 (a), BR-4 responds in a similar manner. The system immediately detects an inconsistency concerning the spin values for neutron decay and attempts to correct it. (The current program does not address the issue of energy conservation.) BR-4 cannot modify the spin counts of the particles in the reaction, as these values are marked as having been established by observation. This leaves revision of the unbalanced reaction as the only solution.

One such revision adds an extra particle to the output side of the reaction, giving $n \rightarrow p + e + \nu$. Using the conservation laws as constraints, the system computes the mass, charge, and spin of the new particle, ν, as 0.0, 0, and $\frac{1}{2}$, respectively. Another possible revision would have added a new particle with the opposite spin to the input side of the reaction. However, we believe physicists favored the former solution because they were thinking in terms of a decay process, so we have biased BR-4 in this direction as well.

Our treatment of this episode ignores many details, including the role that conservation of energy, in addition to spin, played in driving proposals for the neutrino. But the general line of reasoning, that a new particle with certain quantum values was needed to preserve conservation, appears historically accurate, and BR-4's heuristics arrive at the same description for this particle as did Fermi and his colleagues.

3.2 Proposing the Baryon Number

The inference of the neutrino left physicists with six elementary particles, having the properties and values shown in Table 1 (a) and (b). Scientists realized that the existence of these particles, combined with the existing conservation laws, implied a variety of reactions. Subsequent experiments revealed evidence for some of these reactions, shown in Table 2 (a), but not for some others, shown in Table 2 (b). For some reason, the three predicted decays of the proton did not occur in nature; to remedy this problem, physicists proposed a new quantum property, known now as the *baryon number*.

Given the six particles in Table 1, our model follows a similar line of reasoning. BR-4 realizes that its current theory is incomplete, so it predicts all decay and collision reactions involving these entities (up to length three) that conserve

Table 3. The quantum values of particles known in 1953, after discovery of baryon and lepton numbers.

Particle	mass	charge	spin	baryon	lepton
γ	0.00	0	1	0	0
e	0.51	-1	$\frac{1}{2}$	0	1
p	938.26	1	$\frac{1}{2}$	1	0
n	939.55	0	$\frac{1}{2}$	1	0
$\bar{e}$	0.51	1	$\frac{1}{2}$	0	-1
ν	0.00	0	$\frac{1}{2}$	0	1
μ	105.60	-1	$\frac{1}{2}$	0	-1
$\bar{\mu}$	105.60	1	$\frac{1}{2}$	0	1
π	139.60	1	0	0	0
$\bar{\pi}$	139.60	-1	0	0	0
π_0	135.00	0	0	0	0

charge and spin, giving the seven reactions[2] in Table 2. These correspond to proposed experiments with the particles, or at least to suggestions for what to look for in such experiments. When informed that the reactions in Table 2 (a) occur but those in (b) do not, BR-4 infers that its theory is incomplete in a deeper sense and proposes a new property to correct the situation.

To determine the values of this new property, BR-4 selects one of the missing reactions, say $p \rightarrow \bar{e} + \gamma$, and turns it into a set of inequalities, each based on a different combination of values for the particles involved. In this case, it generates the four inequalities $1 \neq 0+0$, $1 \neq 1+1$, $0 \neq 1+0$, and $0 \neq 0+1$. The system then selects one of these value sets, say the first, $\{p = 1, \bar{e} = 0, \gamma = 0\}$, and inserts them into one of the observed reactions, say $n \rightarrow p+e+\nu$, this time treating it as an equality.

In this case, BR-4 obtains the expression $n = 1 + 0 + \nu$, which leaves the property values for n and ν unspecified. Two consistent value sets are possible for this pair, $\{n = 1, \nu = 0\}$ and $\{n = 0, \nu = -1\}$. BR-4 selects the first and uses it to check the observed reactions, introducing values for the remaining unassigned particles as necessary. Detection of an unbalanced reaction that violates conservation of the new property causes backtracking to one of the alternative value sets. If the search exhausts all such sets produced from the observed reactions, the system backtracks further and considers other value sets generated from the unobserved reactions.

[2] BR-4 also generates two other reactions, besides $n \rightarrow p+e+\nu$, that involve neutrinos: $\nu+p \rightarrow n+\bar{e}$ and $\nu+n \rightarrow p+e$. However, physicists showed little concern when they did not immediately detect these reactions, presumably because theory predicted that neutrinos interacted very rarely. Thus, we told the system to ignore them at this stage of our simulation.

Table 4. Some particle reactions that were (a) observed and (b) not observed in experiments after the discovery of mesons.

(a) Observed reactions	(b) Unobserved reactions
$\mu \rightarrow e + \nu + \nu$	$\mu \rightarrow e + \gamma$
$\pi \rightarrow \bar{e} + \nu$	$\mu \rightarrow e + \bar{e} + e$
$\pi \rightarrow \bar{\mu} + \nu$	$\pi_0 \rightarrow e + \bar{\mu}$
$\pi \rightarrow \pi_0 + \bar{e} + \nu$	$\pi_0 \rightarrow \mu + \bar{e}$
$\pi_0 \rightarrow \bar{e} + e$	
$\pi_0 \rightarrow \nu + \nu$	
$\pi_0 \rightarrow \gamma + \gamma$	

Given the experimental results in Table 2, BR-4 arrives at the value zero for all particles except the proton and neutron, to which it assigns the value one, as shown in the first six rows of Table 3. These settings correspond to those obtained by physicists for the baryon number, which successfully explain the absence of the reactions in Table 2 (b), since they fail to conserve this property. As new particles become known, BR-4 assigns baryon values to them as well, using the same search mechanism.

3.3 Mesons and the Lepton Number

In 1935, Yukawa proposed the existence of additional particles in the nucleus, with a mass of about 100 MeV. The reasoning behind Yukawa's proposal, which we have not attempted to model, involved energy calculations on atomic nuclei. Later, in the 1940s, observations of cosmic rays revealed five such particles: the muon (μ) and anti-muon ($\bar{\mu}$), the pion (π) and anti-pion ($\bar{\pi}$), and the pion zero (π_0). These suggested a variety of reactions, some that were observed by scientists and others that were not.

Konopinski and Mahmoud (1953) attempted to explain the mismatch between theory and data, focusing on the five detected reactions $\mu \rightarrow e + \nu + \nu$, $\mu + \nu \rightarrow e + \nu$, $p + \mu \rightarrow n + \nu$, $\nu + n \rightarrow p + \mu$, and $\nu + n \rightarrow p + e$ and on the single unobserved reaction $\mu \rightarrow e + \gamma$. In order to explain the absence of this decay, they proposed a new quantum property, the *lepton* number, with nonzero values for the muon, the electron, the neutrino, and their antiparticles.[3] However, Konopinski and Mahmoud assumed that the muon in the reactions was an antiparticle, which led them to assign it the lepton value -1. With the introduction of the lepton number, physicists had produced a theory, equivalent to that depicted in Table 3, that appeared consistent and complete. Many scientists had reservations about Konopinski and Mahmoud's theory, but it was the best account available at the time.

[3] Pais (1986) claims that he suggested the lepton number, including its name, earlier, in 1947, based on an analogy with the baryon number for heavier particles.

Table 5. Particle reactions that were (a) observed and (b) not observed in experiments after distinguishing between electron neutrinos (ν_e) and muon neutrinos (ν_μ).

(a) Observed reactions	(b) Unobserved reactions
$\mu \rightarrow e + \bar{\nu}_e + \nu_\mu$	
$\mu \rightarrow \bar{e} + \nu_e + \bar{\nu}_\mu$	$\bar{\nu}_\mu + p \rightarrow n + \bar{e}$
$\pi \rightarrow \bar{e} + \nu_\mu$	$\nu_\mu + n \rightarrow p + e$
$\pi \rightarrow \bar{\mu} + \nu_\mu$	
$\pi \rightarrow \mu + \bar{\nu}_\mu$	
$\pi \rightarrow \pi_0 + \bar{e} + \nu_\mu$	
$\pi_0 \rightarrow \bar{e} + e$	
$\pi_0 \rightarrow \nu_\mu + \nu_\mu$	
$\pi_0 \rightarrow \gamma + \gamma$	

BR-4 responds to the introduction of mesons in a similar manner. Given the five new particles, it predicts a variety of reactions, including four muon decays, five pion decays, and ten reactions that involve the pion-zero. Table 4 shows a sample of these predictions, some (a) that were observed and others (b) that were not. These differ somewhat from the ones addressed by Konopinski and Mahmoud, who presumably did not mention the observed decays that had been known since 1947 (Griffiths, 1987, p. 19, p. 25) and may have ignored some unobserved ones because the values for the lepton number forbid them.

Upon finding that the predicted reaction $\mu \rightarrow e + \gamma$ has not been observed, BR-4 attempts to introduce a new property with values that rule out this interaction. However, the system cannot find a consistent set of values for this property if, as usual, it considers only zero and positive values. For BR-4 to follow Konopinski and Mahmoud's reasoning, we must tell it (as the physicists concluded) that μ is an anti-particle, which lets the system consider negative quantum values. Table 3 shows the values generated by the system when given this assistance; they correspond to those inferred by Konopinski and Mahmoud, with the exception that μ and $\bar{\mu}$ are reversed.

3.4 Electron and Muon Numbers

In the year 1953, another important development took place. Additional experiments revealed indirect evidence for the predicted reaction $\nu + p \rightarrow n + \bar{e}$, which obeyed all known conservation laws and thus was required for the theory to be complete. Yet this reaction occurred when the neutrino (ν) had been generated through beta decay ($n \rightarrow p + e + \nu$), but not when produced through muon decay ($\bar{\mu} \rightarrow \bar{e} + \nu + \nu$).

To resolve this dilemma, scientists postulated that the two reactions actually generated two distinct types of neutrinos, calling the former an electron neutrino (ν_e) and the latter a muon neutrino (ν_μ). This distinction (and the analogous

one for anti-neutrinos) introduced two additional rows in the table of particles. However, it also produced the unobserved reactions shown in Table 5 (b), which physicists sought to explain by introducing yet another property and which they named the *electron number*.

Our model cannot directly explain the historical distinction into two classes of neutrinos, but we believe it constitutes a variation on the heuristic for postulating new particles that originally led to inference of the neutrino. The situation also bears some similarity to the distinction inferred by Mendel 1865 to explain the different offspring of apparently identical peas, which Shen and Simon (1989) have modeled using a related mechanism. Langley et al. (1987) have used a similar technique to explain distinctions that occurred in the history of chemistry.

Once this difference has been introduced manually, BR-4 realizes that its current theory is incomplete, in that it cannot explain the unobserved reactions. Postulating a new property, it searches the space of values using the same process as it used for the baryon and lepton numbers. The resulting values agree with those proposed by physicists for the electron number, and they are sufficient to rule out the two unobserved muon reactions shown in Table 5 (b). Physicists also postulated yet another quantum property, called the *muon number*, on grounds of symmetry between electrons and muons. However, lacking any heuristics of this sort, BR-4 cannot reproduce this step in the human scientists' reasoning.

3.5 BR-4 as a Historical Model

We have implemented BR-4's operators and heuristics in PROLOG, and we have verified the system's ability to reproduce the historical discoveries reported earlier. In each case, we gave the system a set of particles, a set of known quantum properties, the hypothesized values for those properties, and a set of observed and unobserved reactions; in response, BR-4 generated the revised values, new particles and properties, and predicted reactions we have described. These formed a partial basis for the next inputs to the system, giving historical continuity to the model's behavior.

The resulting chain of reasoning carries BR-4 through more than two decades of major discoveries in particle physics. Moreover, the system relies on mechanisms that are consistent with our knowledge about the nature of human cognition. In particular, it carries out a limited heuristic search through a space of models that is guided both by knowledge about the domain and by observations. Moreover, this process occurs in an incremental fashion, with the system revising previous models as new phenomena become available and with new results becoming background knowledge for the next round of discovery.

As we have noted, BR-4 does not explain all of the major events in particle physics, even during the period we have attempted to simulate. In a number of cases, we had to intervene manually at selected points beyond the insertion of information about the outcomes of predictions. These steps included telling the system to ignore some unobserved reactions involving neutrinos, to assume that the muon is an antiparticle with nonpositive quantum numbers, and introducing the distinction between electron and muon neutrinos. Also, the system explains

the historical sequence of events at a quite abstract level that ignores many details which occupied particle physicists' time and energy.

Thus, although BR-4 has let us model an extended period in the history of science, it remains an incomplete account. Each situation that required intervention suggests the need for additional mechanisms that should let its successor better match the historical record. These should include heuristics for ignoring predictions that are too difficult to observe, for considering wider ranges of quantum values, and for discriminating particles that appear the same but behave differently. Each such extension seem as general, at least in principal, as the existing operators and heuristics on which BR-4 relies.

4 Related Work on Computational Scientific Discovery

Our computational model of discovery draws many of its ideas from earlier work in this area. BR-4 is a direct descendant of Żytkow and Simon's (1986) STAHL, which modeled a variety of qualitative discoveries in the history of chemistry. The detection of inconsistencies in reactions played an important role in this system, with one of its responses being the introduction of new elements like phlogiston, which served much the same role in early chemistry as the neutrino did in particle physics.

Rose and Langley (1986) described STAHLp, a rational reconstruction of the earlier system that showed all of its discoveries could be explained in terms of inconsistencies and their resolution. In addition, they used STAHLp and REVOLVER (Rose & Langley, 1988), a similar system, to model a number of other reaction-oriented discoveries from the history of science, including some from particle physics. Moreover, their approach showed that dependency-directed reasoning simplified the theory-revision process, letting their systems handle problems with a search-control scheme that relied on incremental hill climbing rather than more systematic search.

Kocabas' (1991) BR-3 system extended this framework to include the detection of incomplete theories and the postulation of new properties to explain the absence of reactions. He applied this idea to the history of particle physics, using it to explain the origin of several quantum numbers and the particular values assigned to them. In related work, Kocabas (1992) adapted similar methods to discovery in the area of superconductivity. BR-3 was the immediate precursor of BR-4, with the former differing mainly in that it lacked the ability to postulate new particles and to predict new reactions.

Valdés-Pérez (1994) has described an alternative approach to discovery in particle physics, which he implemented in his PAULI system. This scheme uses a variation on linear programming to search the space of property values, subject to constraints that reflect observed and unobserved reactions. In addition, Fischer and Żytkow (1992) have reported on GELL-MANN, a system designed to explain the formation of the quark theory, which also carries out a form of constraint-satisfaction search to determine parameter values. Both systems have generated interesting models that differ from those found by human scientists, but these

results, combined with their more powerful and nonincremental search methods, make them less plausible as historical accounts than the STAHL, STAHLp, BR-3, and BR-4 systems.

Despite their differences, each of these systems fits nicely within the framework proposed by Valdés-Pérez, Simon, and Żytkow (1993), which characterizes the discovery process in terms of operations on two related matrices. The various programs differ in their operators for altering the matrices, with BR-3 and BR-4 adding steps for introducing a property, predicting reactions, and positing a particle. PAULI and GELL-MANN also explore a matrix space but invoke different search regimens for selecting operators.

Other research on scientific theory revision, such as Rajamoney's (1990) work on theory-guided experiment generation in physics, seems less closely related. However, Kulkarni and Simon's (1990) KEKADA integrates theory revision, experiment design, and problem formulation to model Krebs' discovery of the urea cycle. The system includes heuristics for making predictions, redirecting attention when they are violated, and designing experiments to determine the underlying cause. The KEKADA work comes the closest to our own in spirit, in that both involve modeling an extended period in the history of science, rather than isolated events. However, Kulkarni and Simon's model operates at a finer granularity and better matches the historical details than does BR-4.

5 Directions for Future Research

Although BR-4 provides an abstract account for some important developments in particle physics, there remains considerable room for improvement. One problem is that the model's coverage of the historical process remains far from continuous. A more complete account would incorporate knowledge about the difficulty of detecting some reactions to explain why scientists chose to ignore some unobserved interactions (e.g., those involving neutrinos) while focusing their attention on others (e.g., those concerning proton decay). We should futher reduce reliance on human intervention by adding an operator like the one described by Shen and Simon (1989) that introduces a distinction between particles (e.g., electron and muon neutrinos) based on behavioral differences observed over time. Heuristics for proposing new particles and quantum properties on theoretical grounds would further strengthen the model.

We also hope to extend the system to introduce componential models, which describe particles at one level as combinations of more primitive ones. Langley et al.'s (1987) DALTON took some steps along these lines to explain relations between chemical molecules and elements, but we can incorporate similar methods into BR-4 to explain the origins of the quark theory and its alternatives. The basic task involves explaining why elementary particles with some quantum properties exist while others do not. BR-4's constraints of consistency and completeness seem well suited for this problem, which involves postulating new component particles (quarks), then searching the space of quantum values and their compositions that satisfy certain constraints (such as symmetry) for known particles and that violate these constraints for nonexistent ones.

Finally, although BR-4 implicitly models social aspects of the discovery process by addressing extended periods to which multiple scientists contributed, it accomplishes this at a very abstract level. A more detailed account of social interactions would include explicit communication among particle physicists, with theorists passing on predictions to experimentalists, who in turn report their observations to theorists. An extended model would also support competition in the development of theories to explain new findings and in finding evidence for predicted events. The history of particle physics is rich in examples of such interactions, and we believe that appropriate revisions to BR-4 would let us model at least some of them. To this end, we should assign different facets of the system's domain knowledge to different agents, which would communicate through a common representation; in addition, separate agents would explore different branches when the search process suggests alternative solutions.

6 Concluding Remarks

In this paper we presented BR-4, an integrated model of historical scientific discovery. We examined the system's behavior on four major problems that arose in particle physics, showing that it can replicate important steps in the historical development of this field, some of which were considered major discoveries when first introduced. In particular, BR-4 proposes the existence of the neutrino to avoid violating conservation of spin, it introduces baryon and lepton numbers to explain the absence of reactions involving proton decay, and it postulates electron numbers to rule out unobserved neutrino reactions. The system also finds appropriate quantum values for each particle and predicts the reactions implied by a set of particles and properties.

The BR-4 model achieves these results using simple processes that appear to have considerable generality. The system employs four basic operators for determining the values of a quantum property, creating new properties, positing new particles, and predicting reactions among known particles. Moreover, it uses consistency and completeness constraints to selectively apply these operators, and it incorporates a depth-first control scheme to carry out search when necessary. These activities operate in a continual loop, with incorrect predictions leading to revised models, which then become the starting point for new discoveries. Together, they let BR-4 explain, with occasional aid from its developers, an extended period in the history of particle physics. The simplicity and generality of these mechanisms suggest that we can explain other aspects of scientific discovery in similar terms, and we hope to test that hypothesis in future work.

Acknowledgements

Portions of this paper appeared in the *Working Notes of the AAAI Spring Symposium on Systematic Methods of Scientific Discovery*. The research described herein was supported by Grant No. N00014-94-1-0505 from the Office of Naval Research and by the Nippon Telegraph and Telephone Corporation.

References

Fischer, P., & Żytkow, J. M. (1992). Incremental generation and exploration of hidden structure. *Proceedings of the ML92 Workshop on Machine Discovery* (pp. 103–110). Aberdeen, Scotland.

Giancoli, D. C. (1995). *Physics: Principles with applications* (4th ed.). Englewood Cliffs, NJ: Prentice Hall.

Griffiths, D. (1987). *Introduction to elementary particles*. New York: John Wiley.

Kocabas, S. (1991). Conflict resolution as discovery in particle physics. *Machine Learning*, *6*, 277–309.

Kocabas, S. (1992). Elements of scientific research: Modeling discoveries in oxide superconductivity. *Proceedings of the ML92 Workshop on Machine Discovery* (pp. 63–70).

Konopinski, E. J., & Mahmoud, H. M. (1953). The universal Fermi interaction. *Physical Review*, *92*, 1045–1049.

Kulkarni, D., & Simon, H. A. (1990). Experimentation in machine discovery. In J. Shrager & P. Langley (Eds.), *Computational models of scientific discovery and theory formation*. San Mateo, CA: Morgan Kaufmann.

Langley, P., Simon, H. A., Bradshaw, G. L., & Żytkow, J. M. (1987). *Scientific discovery: Computational explorations of the creative processes*. Cambridge, MA: MIT Press.

Mitchell, F., Sleeman, D., Duffy, J. A., Ingram, M. D., & Young, R. W. (1997) Optical basicity of metallurgical slags: A new computer-based system for data visualisation and analysis. *Ironmaking and Steelmaking*, *24*, 306–320.

Ne'eman, Y., & Kirsh, Y. (1986). *The particle hunters*. Cambridge: Cambridge University Press.

Nordhausen, B., & Langley, P. (1993). An integrated framework for empirical discovery. *Machine Learning*, *12*, 17–47.

Omnes, R. (1970). *Introduction to particle physics* (Tr. by G. Barton). Wiley Interscience.

Pais, A. (1986). *Inward bound*. Oxford: Clarendon Press.

Rajamoney, S. (1990). A computational approach to theory revision. In J. Shrager & P. Langley (Eds.), *Computational models of scientific discovery and theory formation*. San Mateo, CA: Morgan Kaufmann.

Rose, D., & Langley, P. (1986). Chemical discovery as belief revision. *Machine Learning*, *1*, 423–451.

Rose, D., & Langley, P. (1988). A hill-climbing approach to machine discovery. *Proceedings of the Fifth International Conference on Machine Learning* (pp. 367–373). Ann Arbor, MI: Morgan Kaufmann.

Shen, W. M., & Simon, H. A. (1989). Rule creation and rule learning through environmental exploration. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 675–680). Detroit, MI: Morgan Kaufmann.

Trefil, J. S. (1980). *From atoms to quarks*. London: The Athlone Press.

Valdés-Pérez, R. E. (1994). Discovery of conserved properties in particle physics: A comparison of two models. *Machine Learning*, *17*, 47–67.

Valdés-Pérez, R. E. (1995). Machine discovery in chemistry: New results. *Artificial Intelligence*, *74*, 191–201.

Valdés-Pérez, R. E., Żytkow, J. M., & Simon, H. A. (1993). Scientific model building as search in matrix spaces. *Proceedings of the Eleventh National Conference on Artificial Intelligence* (pp. 472–478). Washington, DC: AAAI Press.

Stimulating Discovery*

Ronald N. Kostoff

Office of Naval Research
Arlington, VA 22217

Abstract. Innovation is critical for maintaining competitive advantage in a high tech global economy, especially for organizations or nations that do not possess low cost labor forces. Many studies on innovation attempt to identify endogenous and exogenous variables that impact innovation [7], in order to better understand the environment that promotes innovation. The author's recent efforts have focused on developing processes for enhancing innovation that exploit the transference of information and insights among seemingly disparate disciplines.
The objective of this paper is to describe and demonstrate a hybrid tandem literature-workshop approach to innovation that eliminates the weaknesses but retains the strengths of each component. The literature-based component identifies the technical disciplines related to the central technical theme of interest, the experts in these disciplines, and promising candidate concepts for innovative solutions. These outputs define the agenda and participants for the workshop-based component. An example of this combined approach is presented for the theme of Autonomous Flying Systems. The hybrid approach appears to be an excellent vehicle for enabling innovation. However, it requires substantial time and effort in both phases.

1 Introduction

The process of innovation is of immense social interest and impact, has been studied extensively, and yet remains poorly understood. A critical factor in many instances of innovation is the transfer of information and understanding developed in one or more disciplines to other, perhaps very disparate, disciplines. With the explosion in availability of information, scientists and technologists find it increasingly difficult to remain aware of advances within their own discipline(s), much less in other seemingly unrelated ones. As science and technology become more specialized, the incentives for interdisciplinary research and development are reduced, and this cross-discipline transfer of information becomes more difficult. The author's observation, from examination of many science and technology (S&T) sponsoring agencies and performing organizations and technical journals, is that *strong cross-disciplinary dis-incentives exist at all phases of program/project evolution, including selection, management and execution, review,*

* The views expressed in this paper are those of the author and do not represent the views of the department of the Navy.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 196–213, 2001.

and publication. To overcome cross-discipline transmission barriers, and thereby enhance innovation, systematic methods are required to heighten awareness of experts in one discipline to advances in other disciplines. Most desirable are methods that *incorporate/require cross-disciplinary access as an organic component.*

This paper presents two different, yet complementary, approaches to increase cross-discipline knowledge transfer and provide the framework for enhancing innovation. One is literature-based, the other is workshop-based. Each approach individually represents a major advance in enabling innovation and discovery, and the hybrid of the two approaches provides a synergy that multiplies their combined benefits.

The literature-based approach is summarized first, followed by the workshop-based approach. The advantages of combining the two approaches are then presented. The details of each approach are presented in the appendices.

1.1 Accessing Linked Literatures for Enhancing Innovation-Summary

The first approach searches for relationships between linked, overlapping literatures, and discovers relationships or promising opportunities not obtainable from reading each literature separately. The general theory behind this approach, applied to two separate literatures, is based upon the following considerations [18].

Assume that two literatures with disjoint components can be generated, the first literature AB having a central theme a and sub-themes b, and the second literature BC having a central theme(s) b and sub-themes c. From these combinations, linkages can be generated through the b themes that connect both literatures (e.g., $AB \rightarrow BC$). Those linkages that connect the disjoint components of the two literatures (e.g., the components of AB and BC whose intersection is zero) are candidates for discovery, since the disjoint themes c identified in literature BC could not have been obtained from reading literature AB alone.

Some initial applications of the first approach have been published in the medical literature [18]. One interesting discovery was that dietary eicosapentaenoic acid (theme a from literature AB) can decrease blood viscosity (theme b from both literatures AB and literatures BC) and alleviate symptoms of Raynaud's disease (theme c from literature BC). There was no mention of eicosapentaenoic acid in the Raynaud's disease literature, but the acid was linked to the disease through the blood viscosity themes in both literatures. Subsequent medical experiments confirmed the validity of this literature-based discovery [2]. (A web site [17], overviews the process used to generate this discovery, and contains software that allows the user to experiment with the technique. A 1998 article in The Scientist outlines perceptions of different knowledgeable individuals on Swanson and Smalheiser's general technique [1].

This literature-based discovery approach is in its infancy. Public and private financial support for this technology are minimal. It is an area that seems to have fallen through the cracks. There is essentially one group that is publishing

results of literature-based innovation and discovery in the credible peer-reviewed literature [18,20,19,15,16,17], and two groups that have published concept papers [3,10]. Presently, the approach is not automatic. It requires much thought, expertise, and effort. The author's group is examining different approaches to make the process more systematic, while reducing the manual labor intensity. Given the potential benefits of the literature-based approach for stimulating innovation, it is truly a technology whose time has come.

Appendix A generalizes and expands upon the literature-based approach, using the Database Tomography (DT) techniques and experience developed by the author since 1991 [4,5,8,10,14,13]. It outlines the theory of the expanded approach, the implementation details, and overviews the range of applications possible with this technique.

1.2 Interdisciplinary Workshops for Enhancing Innovation — Summary

The second approach consists of convening workshop(s) of experts from different disciplines focused on specific central themes. The purpose of such a workshop is to achieve multi-discipline synergies and cross-discipline transfers to generate promising research directions for these central themes. The theory behind this approach is described in Appendix B. To test this theory, a workshop on Autonomous Flying Systems was convened in December 1997, and the implementation mechanics and results are described in detail in Appendix B.

The total workshop process consisted of three phases:

(1) a two month pre-meeting e-mail phase in which each participant provided descriptions of advanced capabilities and promising research opportunities from his/her discipline to all other participants;
(2) a two-day meeting at the Office of Naval Research (ONR) during which the promising opportunities identified beforehand were discussed, crystallized, and enhanced; and
(3) a post meeting e-mail phase in which each participant provided additional or embellished opportunities.

A number of important lessons were extracted from the conduct of this workshop, and they can be summarized as follows:

(a) The workshop approach broke new ground toward stimulating innovative thought. It was not easy, simple, or effortless, and required substantial planning and work in order to be effective. One should not throw people from 15 different disciplines together in a room for two days and hope to get new ideas synthesized. There needs to be a common generic thread woven through the different disciplines represented to spark the innovative thought process.

Interdisciplinary workshops, when performed correctly, are the wave of the future in defining new research (and technology) areas and approaches. Because of the intensity and effort involved throughout the process, they are

most appropriate for large scale "grand challenges" in full-blown workshop form, but appropriate as well for smaller scale issues.

(b) Representatives from diverse technical disciplines, organizations, and development categories attended the workshop. There was substantial value in having a balance of discipline, category, and organization diversity at the same meeting. The different perspectives presented benefited all participants.

The use of modern information technology can expand the degree of diversity dramatically. Some of the concepts and group software proposed for network-centric peer review [12] can be easily adapted for use in innovation workshops. This would allow many more people, disciplines, and organizations to be represented, further enhancing the potential for cross-discipline information transfer and resultant innovation and discovery.

(c) Problem selection is crucial. The problem should be sufficiently general that many diverse disciplines can link to it. Given the choice of equally relevant problems, there is more potential for impact in selecting problem areas for which a large interdisciplinary community is not yet obvious.

(d) It is important to select participants by the most objective processes available. A combination of expert recommendation and strategic topical maps based on computational linguistics, publications, and citations was used for the selection process, and this approach produced highly knowledgeable individuals. Incorporation of the full literature-based approach to innovation in the discipline or participant selection process could further enhance confidence that the most appropriate mix of disciplines and experts has been chosen.

(e) It is extremely important that individuals selected for participation be world-class experts in their particular areas. There are relatively very few individuals producing the seminal works in any field [8,9], and it is these people who should be central to any truly innovative workshops. However, in addition to these established experts, highly competent individuals new to the field should also be selected. One benefit of transcending selection of known experts is that fresh faces who are new to established communities appear. They can sometimes challenge established paradigms and offer concepts typically not advanced through panels based solely upon well-known, over-used panelists.

(f) The e-mail component of the workshop is crucial. The gestation period between the input of promising ideas and their actual discussion at the workshop allows consideration of many different approaches and syntheses. It also saves substantial time at the workshop by clarifying confusing issues beforehand. However, in the first experience reported here, the stimulation of dialogue in the e-mail phase among most of the participants did not occur. The only participant to raise questions was the author, and this occurred only a few times. Nonetheless, in these instances, the dialogue was extremely valuable in clarifying issues and surfacing points of contention. In future workshops, it is strongly recommended that a few individuals representing different disciplines be asked to assume a role of facilitator, with the task

of stimulating dialogue and raising questions during the workshop build-up phase.

(g) All the attendees at the workshop were required to participate; there were no pure observers. This meant that they had to submit accomplishments and opportunities statements by e-mail. They also had to be prepared to lead discussions at the workshop. This participation requirement was valuable in that each attendee obtained a sense of ownership in the workshop and its outcome. His/her contribution tended to be more substantive and creative than is typically the case at standard workshops. Those who contributed more in the e-mail phase tended to contribute more in the workshop phase. In addition, there was a sense of equality among participants when all were required to contribute, as opposed to an audience/performer environment with passive onlookers. On the other hand, the downside of requiring attendees to be active participants was that attendance had to be limited. This may not be totally bad, since audience participation can substantially enrich workshop discussion.

(h) In general, there needs to be some incentive to motivate participation of world-class experts in these workshops. Unless they are able to envision some type of substantive impact resulting from their participation, either on larger S&T issues or in their individual disciplines, they could be reluctant to invest the substantial amount of time required for serious participation. This, however, did not turn out to be a problem for the Autonomous Flying Systems workshop, apparently because of the limited size of the field and the interest of the participants in the type of workshop conducted.

In addition, during the workshop, participants did not appear to have reluctance in sharing new concepts. This is in stark contrast to some workshops the author has attended where novel ideas were held very closely. In the Autonomous Flying Systems workshop, there was a spirit of comeraderie and cooperation that pervaded the proceedings, and helped overcome the barriers to sharing. This spirit was fostered in the pre-meeting e-mail dialogue phase, and further nurtured during the meeting by having all attendees participate in the proceedings as equal partners.

Finally, interdisciplinary workshops are a powerful potential source of radically innovative ideas if conducted properly. There are three central requirements for success:

(1) A problem of significant interest to the sponsoring organization must be selected;
(2) An optimal mix of world-class experts appropriate to the problem must be chosen;
(3) Conditions must be created which will motivate the participants to share their novel concepts.

The Autonomous Flying Systems workshop addressed these three requirements to a significant degree. A preliminary concept proposal emerged, and a copy of this proposal is available from the author.

2 Need for Literature/Workshop Synergy

Most organizations use some variant of a workshop/group dynamics approach for brain-storming or other proxies for stimulating innovation. The most current information is available, and real-time information exchange is unmatched. The attendees and participants in these groups tend to be focused subject experts representing a small fraction of the relevant technical community; there is rarely any complementary sophisticated literature analysis performed, and there are rarely experts present from strongly divergent disciplines. The outputs and discussion are highly subjective. The workshop techniques tend not to make full use of many of the information technology advances of recent years. Probably most importantly, there are strong disincentives for the participants to reveal the latest innovations. What many workshops produce in practice are forums for "selling" completed or near-completed efforts.

A few performers, individuals or small groups of individuals, pursue the literature-based computer-assisted approach. This literature approach tends to be more sophisticated and technologically advanced than the workshop approach, and is more objective. It is more comprehensive, since it encompasses S&T beyond the scope of any individual, or group of individuals, and can access data from many technical disciplines and many global sources. The base data is not as current as the workshop approach, due to the documentation time lag. However, with the advent of extensive on-line documentation, this time lag has been reduced considerably. One intrinsic limitation is that a relatively modest amount of S&T performed globally is documented and readily accessible to the wider user community [11]; obviously, any S&T not documented cannot be accessed. The literature-based approach has not received widespread attention and may fall short of the interpretive and analytical strengths of the workshop approach. As a result, the literature approach is not widely used (e.g., [1]).

While either the workshop approach or the literature approach can be done independently to help stimulate discovery, they should be done in tandem to maximize the benefit provided by each. There is nothing on record to indicate that this joint approach to innovation has been implemented, or even considered. The Autonomous Flying Systems workshop described in this paper has some elements of the combined approach. Some of the DT proximity analysis tools were used to identify the scope of related literatures, and the prolific individuals in these literatures. These individuals were then invited to the workshop. However, time constraints precluded using the full capabilities that the literature-based approach can offer.

In a joint workshop-literature effort, the literature approach would be included in the background pre-meeting phase of the workshop approach (as developed in Appendix B). Accordingly, the literature study would provide:

(1) background reading for the workshop participants in related yet disparate science and technology areas;
(2) strategic maps of the broader science and technology literature as outlined in the DT papers referenced above;

(3) promising opportunities for innovation and discovery; and
(4) the disparate science and technology disciplines from which the experts for the workshop could be drawn.

The hybrid literature-workshop approach would eliminate the limitations of each approach done separately. The right people from the right combination of disciplines could be identified by the literature-based approach, and invited to the workshop. The literature-based analysis could structure the technical relationships, and provide an objective starting point for discussion. Network-centric peer review would allow linking, and fusing information from, large numbers of reviewers to incorporate more representative opinion sampling from the larger technical community. The only limitation not overcome is the disincentives for the participants, or document authors, to reveal their latest S&T efforts and innovations.

There is extra time and cost involved with two approaches, and if responses were required with severe time limitations, then only one approach might prove feasible. For organizations that are serious about innovation and discovery, the additional time should not be a factor, given the potential high marginal benefits. Government could probably draw upon a more eclectic group than industry. Because of the competitive aspects, industry would probably rely more upon internal participants and contracted consultants, whereas government would draw upon individuals from many organizations.

3 Conclusions

The advent of large databases, and the parallel advances in computer hardware and software, provide the opportunity to augment and amplify traditional approaches of human creativity in generating discovery and innovation. This paper has shown that multi-discipline structured workshops can enhance the S&T innovation process, and has shown that multi-discipline literature-based analyses can enhance the S&T discovery process. The document has shown conceptually that the combination of computer-enhanced literature-based analyses and multi-discipline structured workshops has the synergistic potential to dramatically improve the discovery and innovation process relative to the already strong capabilities available from each process separately. This literature-workshop synergy represents a potential major breakthrough for systematically identifying: 1) the most promising disciplines to be used in the workshop; 2) specific experts from these different disciplines; 3) candidate promising concepts that form the basis for discussion.

References

1. R. Finn. Program uncovers hidden connections in the literature. *The Scientist*, 11, May 1998.

2. M. D. Gordon and R. K. Lindsay. Toward discovery support systems: A replication, re-examination, and extension of Swanson's work on literature-based discovery of a connection between Raynaud's disease and fish oil. *JASIS*, 47(2), 1996.
3. M. A. Hearst. Untangling text data mining. In *Proceedings of ACL 99, the 37th Annual Meeting of the Association for Computational Linguistics, University of Maryland*, June 20–26 1999.
4. R. N. Kostoff. Database tomography for technical intelligence. *Competitive Intelligence Review*, 4(1), 1993.
5. R. N. Kostoff. Database tomography: Origins and applications. *Competitive Intelligence Review, Special Issue on Technology*, 5(1), 1994.
6. R. N. Kostoff. Database tomography for information retrieval. *Journal of Information Science*, 23(4), 1997.
7. R. N. Kostoff. The handbook of research impact assessment. www.dtic.mil/dtic/kostoff/index.html, 1997.
8. R. N. Kostoff. Database tomography for technical intelligence: A roadmap of the near-earth space science and technology literature. *Information Processing and Management*, 34(1), 1998.
9. R. N. Kostoff. Hypersonic and supersonic flow roadmaps using bibliometrics and database tomography. *Journal of the American Society for Information Science*, Apr. 1999.
10. R. N. Kostoff. Science and technology innovation. In *Technovation*, volume 19, Oct. 1999.
11. R. N. Kostoff. The underpublishing of science and technology results. *The Scientist*, 1, May 2000.
12. R. N. Kostoff. Advanced technology development peer review — a case study. *R&D Management*, 2001. Accepted for publication.
13. R. N. Kostoff, K. A. Green, D. R. Toothman, and J. Humenik. Database tomography applied to an aircraft science and technology investment strategy. *Journal of Aircraft*, 37(4), July-August 2000.
14. R. N. Kostoff, A. S. T. Braun, D. R. Toothman, and J. Humenik. Fullerene roadmaps using bibliometrics and database tomography. *Journal of Chemical Information and Computer Science*, Jan-Feb 2000.
15. N. R. Smalheiser and D. R. Swanson. Assessing a gap in the biomedical literature — Magnesium — deficiency and neurologic disease. *Neurosci Res Commun*, 15(1), 1994.
16. N. R. Smalheiser and D. R. Swanson. Calcium-independent phospholipase A (2) and schizophrenia. *Arch Gen Psychiat*, 55(8), 1998.
17. N. R. Smalheiser and D. R. Swanson. Using ARROWSMITH: A computer assisted approach to formulating and assessing scientific hypotheses. *Comput Meth Prog Bio*, 57(3), 1998.
18. D. R. Swanson. Fish oil, raynauds syndrome, and undiscovered public knowledge. *Perspect Biol Med*, 30(1), 1986.
19. D. R. Swanson. Computer - assisted search for novel implicit connections in text databases. In *Abstr Pap Am Chem S 217*, 1999.
20. D. R. Swanson and N. R. Smalheiser. An interactive system for finding complementary literatures: A stimulus to scientific discovery. *Artif. Intell.*, 91(2), 1997.

A Literature Approach

A.1 Overview

The theoretical basis of the literature approach mirrors the scientific process in many ways. Information from diverse literatures, with relevant interfaces, is examined. All information is first analyzed and then synthesized to produce discovery and innovation. Initial work [18,2] examined three variable classes or themes (c, b, a) in two literature categories (C and B) using two different approaches (start with c, determine b, then determine a; start with c and a, then determine b).

(**NOTE:** *The sequence abc will typically (but not always) represent a time-varying process, such as a procession from research to development to systems. Where this sequence does represent a temporal process, the convention used in the remainder of this appendix is that the alphabetical designation of variables follows the arrow of time. Thus, a might represent a research variable or theme, b might represent a technology variable or theme, and c might represent a system variable or theme. The terms "variable" and "theme" are treated as interchangeable; "thematic variable" is used in places to emphasize this congruence.*)

The principal thematic variables determine a thematic literature. From the previous example, if Raynaud's disease is the thematic variable specified initially, then the corresponding thematic literature might be all the papers in a given database that contain the phrase Raynaud's disease. The remaining thematic variables and literatures are determined by applying different algorithms to the initial thematic literature and subsequent derived literatures. Again, from the previous example, an algorithm would be applied to the Raynaud's disease thematic literature to determine the thematic variable blood viscosity, and a derived literature could then be determined as all the papers in a given database that contain the phrase 'blood viscosity'.

The first approach in the initial reported work [18,2] could be viewed as addressing the question: What variables a could influence variable c through mechanisms b, or, in the example described above, "What treatment factors a could influence Raynaud's disease c through the different mechanisms b." This approach started with thematic variable c (e.g., Raynaud's disease), and used this variable to develop thematic literature C. Algorithms were applied to this thematic literature database to identify thematic variable b values (b_1, b_2, etc., representing characteristics such as blood viscosity, blood flow, blood platelets, poor circulation, and others) closely linked to thematic variable c. Each value or theme of variable b (b_1, b_2, etc.) was used to develop a thematic literature B_1, B_2, etc. Algorithms were applied to each of the thematic B literatures to identify thematic variable a values (a_1, a_2, etc. representing characteristics such as fish oil, eicosapentaenoic acid, and others) closely linked to the specific thematic variable b of each thematic B literature. Values of the thematic a variables in each of the thematic B literatures not found in thematic literature C defined a subset of the thematic B literatures that was disjoint from thematic literature C (e.g., the term "fish oil" was not found in the Raynaud's disease literature).

These disjoint thematic a variables and their associated thematic B literature subsets became candidates for discovery and innovation.

The other approach reported could be viewed as addressing the question: What are the mechanisms b through which variable a could impact variable c. This approach started with variables c and a, and their associated literatures C and A, and identified variables b that were linked to both variables c and a. The same types of algorithms as in the first approach were used to identify closely linked variables, and the requirement for disjointness between literatures C and A was used as a basis for discovery.

From the experience of these two approaches, it becomes clear that the independent and dependent variables chosen, and the algorithmic approach selected, depend on the question being asked. Further examination shows that other approaches beyond these two are possible to answer other questions. The present paper examines seven approaches to generate innovation and discovery that are structured to answer seven different questions, and shows how the algorithms and techniques developed in Database Tomography are used in these approaches. More specific computational details of the latter six approaches approach can be found in [10].

A.2 Specific Approaches

The following discussion will be limited to scenarios of three variables a, b, c, and two literatures. In future studies, more complex cases could be candidates for analysis and experimentation.

For the simple two literature/three variable case, seven separate generic cases are possible, where the variables specified can be viewed as "independent" and the variables determined can be viewed as "dependent:"

(1) specify a, determine b and c; (2) specify c, determine a and b;
(3) specify b, determine a and c; (4) specify a and c, determine b;
(5) specify a and b, determine c; (6) specify b and c, determine a;
(7) specify a and b and c, validate linkage existence.

Cases (1), (2), and (3) are the most open-ended and least constrained. In each case, one variable is specified, and the other two are determined using the DT algorithms, the condition of disjointness and, most importantly, expert judgement. Cases (4), (5), and (6) are more constrained, since two variables are specified, and the third is determined using similar processes to the above. Case (7) is fully constrained, and its purpose is to ascertain literature support for validation of a hypothetical relation between specified values of the three variables. Cases (4) and (5) are subsets of case (1); cases (4) and (6) are subsets of case (2); cases (5) and (6) are subsets of case (3); Case (7) is a subset of cases (1) through (6). The solution mechanics for each of these seven cases will now be outlined.

Opportunity Driven. This first case addresses the question, "What are the potential variable c impacts that could result from variable a, and what are the

variable b mechanisms through which these impacts occur?" One specific variant of this question is of particular interest and importance to the science and technology community, "What are the potential impacts on research, development, systems, and operations that could result from research on a given topic?"

If the generic question of this first case is applied to the above example for the case where variable a is "fish oil" only, it could be phrased as, "What are the potential impacts or benefits (positive or negative) resulting from fish oil that would not be obvious from examining the fish oil literature alone?" This is an open-ended question, and places no restrictions on the mechanisms b or the types of impact c. The first case is represented schematically as: $a \rightarrow b \rightarrow c$.

Here, a is the independent variable, and b and c are the dependent variables that result from the solution process. The operational sequence is to start with the variable a and generate a literature A. Again following the above example and using the abbreviations FO (fish oil), BV (blood viscosity), and RD (Raynaud's disease), this means that the process would start by identifying the FO literature (call this A_1). Many approaches could be used to define this literature; the approach recommended here is the one used in recent DT studies [14,13] for defining literatures. As an example of one literature definition approach, the iterative Simulated Nucleation method [6] would be used to identify all the papers in the Science Citation Index (SCI) which contained FO (and other related terms in the query) in the title, keywords, and abstract fields. This collection of papers would constitute the FO literature

(**NOTE:** *Use of the SCI is one example only. Because DT uses full text databases, there is no limitation in any database selected to titles or key works or index words, and many different types of databases or free text can be used for the analysis.*)

The next step in the process is be to identify the variables b $(b_1, b_2, \ldots)$ linked closely to variable a_1, and then identify the literatures B associated with variable b $(B_1, B_2, \ldots$.the BV literatures). For this step, the proximity analysis method used in the recent DT studies (or other co-occurrence techniques) would be employed. For a journal based database, this method conceptually identifies phrases in paper titles or abstracts or main texts physically located near the term of interest. As an example, if the term of interest in a given database is Raynaud's disease, then the proximity analysis method would provide a list of all phrases in close physical proximity to the term Raynaud's disease for all occurrences of this term in the text. The proximity analysis approach of DT is based on the experimental findings that phrases within a semantic boundary (same sentence, paragraph, etc.) located physically close to the term of interest are contextually and conceptually close to the term of interest. Continuing the above example, this step uses the proximity analysis of DT to identify phrases in the FO literature physically close to the term FO, such as b_1, b_2, etc.

For each of these identified phrases b_1, b_2, etc. , a literature $(B_1, B_2, \ldots)$ is established by querying the SCI. The next step is, for each of these B literatures, to identify the linked variables c $(c_1, c_2, \ldots)$ The process used to identify the variables b_1, b_2, etc. linked to variable a_1 is repeated to obtain the variables

c_1, c_2, etc. linked to each value of variable b. The subsets of the B literatures which are disjoint from literature A_1 (e.g., the B literatures which don't contain the term FO) must then be identified, and the variables c (and their associated linking mechanisms b to variable a_1) within these disjoint B literature subsets then become the candidates for discovery and innovation.

It is obvious that the process can easily mushroom out of control unless stringent limiting constraints are placed on the number of B literatures and c variables selected. For example, suppose that three b variables b_1, b_2, b_3 (and their associated three B literatures (B_1, B_2, B_3) are identified as closely linked to FO. Suppose also that each of these three b variables is closely linked to five c variables. Then four literature searches are required (A_1, B_1, B_2, B_3), and fifteen abc linked pathways must be examined for disjointness and discovery, according to the following:

$$a_1 \to b_1 \to c_{11};\ a_1 \to b_1 \to c_{12};\ a_1 \to b_1 \to c_{13};\ a_1 \to b_1 \to c_{14};\ a_1 \to b_1 \to c_{15};$$
$$a_1 \to b_2 \to c_{21};\ a_1 \to b_2 \to c_{22};\ a_1 \to b_2 \to c_{23};\ a_1 \to b_2 \to c_{24};\ a_1 \to b_2 \to c_{25};$$
$$a_1 \to b_3 \to c_{31};\ a_1 \to b_3 \to c_{32};\ a_1 \to b_3 \to c_{33};\ a_1 \to b_3 \to c_{34};\ a_1 \to b_3 \to c_{35};$$

In reality, there will be hundreds, if not thousands, of candidate b and c variables. However, there are different ways by which the b and c variables can be sharply limited in number. First, the analysts performing the study would eliminate all non-technical content phrases that passed through the trivial word filter in the DT algorithm. Second, the numerical indices for each phrase generated by the DT proximity algorithm would be used as one figure of merit for preselection of key phrases. Third, those c variables that reappear in different abc pathways would have a higher priority for selection. Fourth, analyst judgement would be applied to weight the potential value of the different abc pathways in computing figures of merit.

The literature searches and proximity analyses are fairly straightforward, and have been refined in the DT process. The main intellectual efforts must be focused on prioritizing and reducing the number of linked variables or literatures to be examined, and interpreting the relationships among the final disjoint literatures to generate potential discovery relationships.

Requirements Driven. This second case addresses the question, "What are the variables a that could impact variable c, and what are the variable b mechanisms by which these impacts are produced?" Applied to the above example for the case where c is Raynaud's disease only, it could be phrased as "What are the factors and their associated mechanisms that could impact the course of Raynaud's disease that would not be obvious from examining the Raynaud's disease literature alone?" This second case is represented schematically as: $a \leftarrow b \leftarrow c$. Here, c is the independent variable, and b and a become the dependent variables.

Mechanism Driven. The third case addresses the question, "For a given mechanism b, what are the variables a that could impact the variables c?" Applied

to the above example for the case where b is blood viscosity, it could be phrased as, "What combinations of variables that could effect a change in the blood viscosity mechanism and could be impacted by a change in the blood viscosity mechanism are candidates for discovery that were not obvious from examining only the blood viscosity literature?" The third case is represented schematically as: $a \leftarrow b \rightarrow c$. Here, b is the independent variable, and a and c are dependent variables.

Opportunity-Requirements Driven. This fourth case addresses the question, "What are the mechanisms b through which variable a could impact variable c?" Applied to the above example for the case where c is Raynaud's disease only, and a is fish oil only, it could be phrased as, "What are the mechanisms through which fish oil could impact Raynaud's disease that would not be obvious from examining only the Raynaud's disease literature or the fish oil literature?" The fourth case is represented schematically as: $a \rightarrow b \leftarrow c$. Here, variables a and c are independent, and variable b is the dependent variable.

Opportunity-Mechanism Driven. The fifth case addresses the question, "What are the variables c which could be impacted by variable a through mechanism b?" Applied to the above example for the case where b is blood viscosity only, and a is fish oil only, it could be phrased as, "What abnormalities could be influenced from the impact of fish oil on blood viscosity that would not be obvious from examining only the abnormality's literature or the fish oil literature?" The fifth case is represented schematically as: $a \rightarrow b \rightarrow c$. Here, a and b are the independent variables, and c is the dependent variable.

Requirements-Mechanism Driven. The sixth case addresses the question, "What are the variables a that could impact variable c through mechanism b?" Applied to the above example for the case where b is blood viscosity only, and a is fish oil only, it could be phrased as, "What factors could impact Raynaud's disease by impacting blood viscosity that would not be obvious from examining only the factors' literature or the Raynaud's disease literature?" The sixth approach is represented schematically as: $a \leftarrow b \leftarrow c$. Here, b and c are the independent variables, and a is the dependent variable.

Opportunity-Mechanism-Requirements Validation. The seventh case addresses the question, "Does the literature support the possibility that variable a could impact variable c through mechanism b?" Applied to the above example for the case where a is fish oil only, b is blood viscosity only, and c is Raynaud's disease only, it could be phrased as, "Does the literature support the possibility that fish oil could impact Raynaud's Disease by altering blood viscosity in a way that would not be obvious from examining only the fish oil literature or the Raynaud's disease literature?" The seventh approach is represented schematically as: $a \leftrightarrow b \leftrightarrow c$. Here, a and b and c are independent variables.

B Crossing the Bridge: Interdisciplinary Workshops for Innovation

B.1 Background

ONR established a series of workshops in 1997 aimed at promoting innovation while also enhancing organization, category, and discipline diversity components. The focus of the first novel workshop founded on this plan was "Autonomous Flying Systems," an area of perceived long-term interest to not only the Navy and DOD, but also to NASA and other governmental and industrial organizations. The process employed was designed starting with a clean slate and was intended for application to very significant technical challenges. The present appendix further describes the process that was used to identify the technical theme of the workshop, select the participants, and conduct all three phases of the total workshop.

B.2 Workshop Theme Identification

It was decided that the initial workshop theme should 1) focus on problems related to the main science and technology emphasis area of the author's home organization, Strike Technology, and 2) help establish the most supportive environment for innovation. The problem selected should be focused and understandable, and it should have a generic technical base amenable to soliciting people from many different disciplines. The topic finally selected was autonomous control of unmanned air vehicles, including takeoff and landing from limited areas on smaller Navy ships. It was apparent that the underlying science and technology permeated many different disciplines, including aerodynamics, controls, structures, communications, guidance, navigation, propulsion, sensing, and systems integration. Also, the naval applications for some aspects of this problem were sufficiently unique that probably not a great deal of work had been done in this area. Subsequent literature analyses validated this assumption.

Present naval air systems are either manned (most aircraft) or tele-operated, semi-autonomous (weapons and some aircraft). The weapons are a mix ranging from "dumb" bombs and shells to "smart" missiles. The future trend is toward "smart" autonomous or semiautonomous aircraft and weapons. Since a major role of ONR is to proactively address the technology that will influence future naval forces, it seemed natural to examine S&T roadblocks on the path to unmanned autonomous "smart" flight systems. Consequently, the focus of the initial workshop was defined as identification of the fundamental operational principles of autonomous flying systems over a fairly wide range of flight environments. In particular, the workshop was aimed at examining what had been learned about autonomous or semiautonomous operation from the animal (mainly flying) kingdom and from other unmanned autonomous/semiautonomous tele-operated systems such as autonomous underwater vehicles and locomoted robots. Animals are now being studied as integrated systems by scientists on the forefront

of biological research. The issues of aerodynamics, flight mechanics, dynamic reconfiguration, materials, control, neuro-sciences, and locomotion are not being studied as separate disciplines by these scientists, but rather are being studied in parallel in the same animal system and in their relation to the function and mission of the animal system. While this integrative biological research is in its infancy, and results are only starting to emerge, the time seemed appropriate for assembling these diverse groups and exploiting their synergy. Not only could there be benefit to the Navy from such cross-discipline interaction, but benefit could be possible for each of the contributing disciplines as well.

A major thrust of the workshop was projected to be identification of the autonomous operational principles for each unique system and the relation of these principles to mission and function, then extraction of the generic operational principles that underlay all the systems, both biological and man-made. It was hoped that the cross fertilization of disciplines would be able to further elucidate and clarify the more important generic concepts, and then provide insight that could be utilized to enhance the autonomous operation of naval flying systems.

B.3 Participant Selection

Once the theme of the workshop was established, a sub-theme taxonomy was developed to focus the agenda and to identify workshop participants. A dual approach was followed to generate the taxonomy.

Discussions were held with agency experts on the generic theme concerning the taxonomy structure. In parallel, the SCI was queried for papers related to the generic theme. Both bibliometric and computational linguistics analyses of these papers were performed to provide strategic maps of the topical area, identifying key performers, journals, institutions, and their relations to the technical themes and sub-themes of the workshop. A taxonomy was constructed based on these strategic maps. (For a description of how the bibliometric and computational analyses are combined to generate strategic maps, see [8,10]).

(**NOTE:** *If a combined literature-based and workshop-based approach is taken for stimulating innovation, as recommended in this paper, the literature-based analysis would be done at this stage.*)

Both of these taxonomy sources, in-house experts and the SCI, then provided initial candidates for participation in the workshop. These candidates were contacted, and asked to suggest additional candidates. This procedure continued until a large pool of potential candidates was established. Three main selection criteria for workshop participants were established;

(1) multiple recommendations,
(2) significant publications is the field, and
(3) literature citations.

These three criteria were tempered with judgement to insure that bright young individuals, who had not yet established a track record, were not excluded

from the pool, and that the panel as a whole had the correct level of discipline, category, and organization balance. In addition, a guideline was established that all workshop attendees would be active participants, so the number of attendees was limited to facilitate discussion and interactions.

All these constraints, guidelines, and selection criteria were used to arrive at the final panel size and structure. The result was a panel of slightly more than twenty people representing a mix of disciplines that included biologists (experts in bird, bat, frog, fish, or insect studies), robotics, artificial intelligence, controls, autonomous aircraft, fluid dynamics, sensors, neuroscience, cognitive science, autonomous underwater vehicles, aerodynamics, propulsion, and avionics.

B.4 Overview of Workshop Process Steps

Workshop Buildup. The buildup period for the Workshop in question started about two months before the meeting. Specific guidance for the conduct of the workshop was sent to the participants by e-mail, including a statement of the naval technical problems to be addressed. The technical component of the buildup phase was then conducted by e-mail.

The main purpose of this buildup phase technical component was to have each participant generate new ideas from his/her discipline for all other participants to consider. The other participants could then dialogue by e-mail to clarify/modify/embellish these ideas. At a minimum, even if no dialogue resulted, there would be a gestation period of about two months for each participant to absorb these concepts from other disciplines. Specifically, each participant was requested to:

- submit a half dozen leading edge capabilities or accomplishments in his/her discipline(s) that could potentially impact the naval technical problems; and
- identify several leading edge capabilities or accomplishments projected in his/her discipline(s) over the next decade that could potentially influence the naval technical problems; and
- submit a few leading edge capabilities or accomplishments in his/her discipline(s) whose impact on the naval technical problems was not obvious to him/her, but might be obvious to someone else.

The participants were free to comment on potential relations among any of the capabilities, accomplishments, or combinations of capabilities and accomplishments, and any of the naval technical problems, or combinations of problems. One of the functions of the participants from the author's organization was to facilitate and stimulate dialog by raising questions and issues on the submitted information.

In actual practice, most of the comments generated resulted from questions stimulated by the discussion facilitator, the author. All of the comments received were then sent to all the participants. This exercise helped stimulate the thinking of the participants, and provided a documented record of the process.

If any of the participants saw a capability or accomplishment from another participant that could impact a problem in his/her discipline, but not impact a

naval technical problem, then the two participants were free to dialog together without informing all the participants. However, these two participants engaged in independent dialog were requested to keep a record of their exchange that might be included with the final workshop report as a potential innovation. This would cover the real possibility of innovation occurring in topics other than the one targeted.

Workshop Meeting. As a result of the ideas presented during the buildup phase, it appeared that the seeds existed for a new S&T program on Autonomous Flying Systems. Therefore, an agenda was sent to the participants with further guidance to address promising S&T opportunities at the workshop, that would serve as the foundation of such a program. Specifically, the participants were asked to address the following issues at the workshop:

- What are the present leading-edge capabilities in your discipline?
- What are the desired future capabilities in your discipline?
- What are the leading research opportunities in your discipline and what additional capabilities could they provide if successful?
- What is the level of risk of these opportunities successfully achieving their targets?
- How would these potentially enhanced capabilities contribute to, or translate into, improved understanding and/or operation of autonomous flying systems?

The meeting occurred on 10–11 December 1997 at ONR. Since some of the leading edge capabilities and potential accomplishments appeared to have applicability to naval technical problems (identified during the e-mail buildup period), the proponent for the capability or accomplishment item took the lead in fleshing out his/her ideas and leading the discussion at the meeting. As a result, the workshop meeting tended to evolve into full panel discussions on each of these potential capabilities.

There were two rounds of discussion at the workshop. The first round consisted of presentations and discussions by each proponent. The second round of the workshop consisted of each participant identifying his/her leading promising research opportunities.

Workshop Cleanup. The participants were requested to provide any additional narrative information that added to or modified their ideas as a result of the workshop experience. The outcomes of the workshop included both the tangible and intangible.

Three immediate tangible outcomes were projected:

(1) A concept proposal for an S&T program focused on Autonomous Flying Systems would be generated;
(2) Technical papers may be submitted to leading science journals based on innovations identified; and

(3) One or more papers on the complete workshop experience might be submitted to leading science journals.

In addition to developing specific topics, it was anticipated that new, unexploited ideas in interdisciplinary research and development might surface during contact between panelists. These novel subjects might form the basis of additional workshops. In addition, extensive lessons were learned as a result of the workshop process. These lessons were summarized in Section 1.2.

Assisting Model-Discovery in Neuroendocrinology

Ashesh Mahidadia and Paul Compton

School of Computer Science and Engineering,
University of New South Wales, Sydney, Australia
{ashesh, compton}@cse.unsw.edu.au

Abstract. It is very difficult, if not impossible, for researchers to manually evaluate and revise their scientific models using a vast amount of relevant information now available to them. The paper describes a new framework, called JustAid, that successfully integrates techniques from Knowledge Acquisition and Machine Learning in a way that complements their strengths to overcome their weaknesses, and provides an interactive environment to help researchers in a process of scientific discovery. JustAid can use information stored in medical databases and assist experimental scientists in forming, testing and revising scientific models, without a need of a knowledge engineer. In this paper, JustAid has been applied to a real world problem in the area of neuroendocrinology, a branch of physiology.

1 Introduction

A web-based service PubMed, of the National Library of Medicine (USA)[1], provides access to over 11 million citations from MEDLINE and additional life science journals. PubMed includes links to many sites providing full text articles and other related resources. The Unified Medical Language System (UMLS) project, also by the National Library of Medicine (USA), develops and distributes multi-purpose, electronic "Knowledge Sources" that can be used by a wide variety of application programs to overcome retrieval problems caused by differences in terminology and the scattering of relevant information across many databases. Thus, systems are being developed to allow researchers to access a vast amount of relevant information more quickly and easily. However, these systems do not yet provide automated tools that can be used to construct, test and invent new scientific models using the available information. This may lead to a possible oversight of important data and insights. This paper describes a new framework and a tool, called **JustAid**, for using information stored in medical databases to assist experimental scientists in forming and revising models.

[1] http://www.nlm.nih.gov/

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 214–227, 2001.

2 Motivation: Hypothesis Testing in the Area of Neuroendocrinology

In this section we introduce a small real world problem from the area of neuroendocrinology. The problem illustrates a type of hypothesis and reasoning used in the area of neuroendocrinology. Smythe carried out experiments to investigate the role of central feedback of glucocorticoids in the control of adrenocorticotropin (ACTH) release by examining the effect of acute dexamethasone pre-treatment on hypothalamic noradrenergic neuronal activity (NNA) and on the secretion of ACTH and corticosterone (CORTICO) in response to stress [5] (see Fig. 1). The author of the paper was interested in explaining the effects of dexamethasone on normal rats and the effects of stress on rats with dexamethasone pre-treatment, using his proposed hypothesis shown in Fig. 1. He carried out a set of experiments to observe the effects of acute dexamethasone pretreatment on hypothalamic noradrenergic neuronal activity (NNA), adrenocorticotropin (ACTH) secreted from the pituitary gland and serum corticosterone (CORTICO) secreted from the adrenals. Importantly he wanted to observe these effects under normal conditions and also in response to stress. He wanted to use stress because it was assumed to activate the whole system. Overall 20 rats were used in the experiment. To induce stress, rats were forced to swim in cold water.

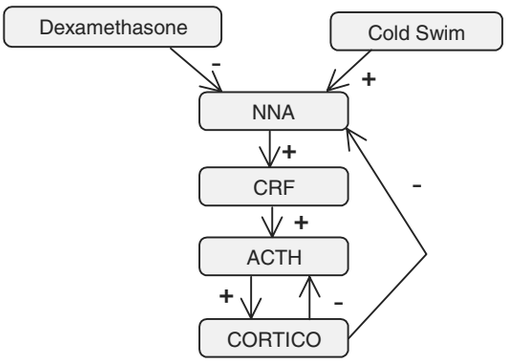


Fig. 1. The hypothesis from [5]

2.1 Experimental Results

The measured parameters in the experiments are NNA, ACTH and CORTICO. The hypothalamic peptides corticotrophine releasing factor (CRF) is not measured in this experiment. Table. 1 shows the values for these parameters after injecting dexamethasone, and also after cold swim stress in rats.

Table 1. Experimental results from [5]

	Experiment-1 controls	Experiment-2 Dexamethasone	Experiment-3 Cold Swim Stress	Experiment-4 Dexamethasone and Cold Swim Stress
NNA (ratio - no units)	0.122	0.105	0.210	0.246
CORTICO (nmol/L)	129	11.3	1232	32.8
ACTH (pg/ml)	89	Undetectable (close to zero)	240	Undetectable (close to zero)

In Table. 1, the second column for "controls" (call it Experiment-1) indicates the values for normal rats (without any treatment). The third column for "Dexamethasone" (call it Experiment-2) indicates the values after injecting dexamethasone. The fourth column for "Cold Swim Stress" (call it Experiment-3) indicates the values after inducing cold swim stress. The fifth column for "Dexamethasone and Cold Swim Stress" (call it Experiment-4) indicates the values after inducing cold swim stress on rats with dexamethasone pre-treatment.

2.2 Hypothesis Is a Causal Model

Fig. 1 shows the hypothesis presented in the research paper [5]. In the hypothesis, links represent causal relations between the nodes (parameters). These relations represent qualitative influences: stimulatory (direct or +) or inhibitory (inverse or -). For example, in Fig. 1, the inhibitory link from CORTICO to NNA represents the following:

increase in NNA can be explained by *decrease* in CORTICO and
decrease in NNA can be explained by *increase* in CORTICO

Similarly, the stimulatory link from ACTH to CORTICO represents the following:

increase in CORTICO can be explained by *increase* in ACTH and
decrease in CORTICO can be explained by *decrease* in ACTH

No further assumptions regarding these links (influences) are made. This is particularly important, as researchers do not have sufficient information regarding such influences, except that they are either stimulatory or inhibitory. For example, they may know that injection of dexamethasone will have an inhibitory effect on hypothalamic noradrenergic neuronal activity (NNA). However, they do not know the precise amount of decrease in NNA for a specific increase in dexamethasone. As discussed in the following section, such hypotheses are used in deriving explanations for experimental results. Fig. 2 shows another (large) hypothesis, from the area of neuroendocrinology, published in [6] and which subsumes the model in Fig. 1.

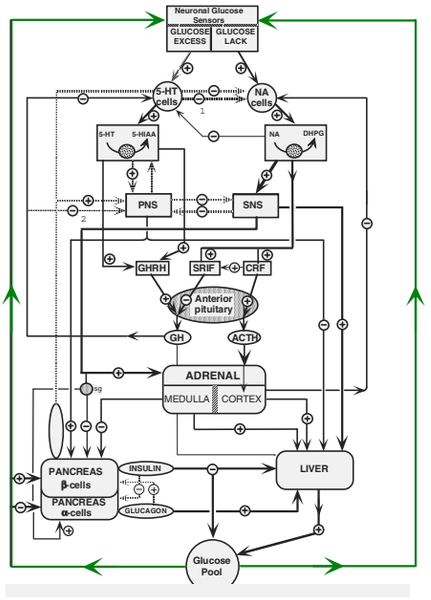


Fig. 2. The hypothesis from [6]

2.3 Completeness of the Model (Explanations for Differences)

As mentioned earlier, the author was interested in examining the effects of dexamethasone (Dex) on normal rats and also the effects of stress on rats with dexamethasone pre-treatment, using his proposed hypothesis (see Fig. 1). The experimental results in Table 1 show that after injecting dexamethasone, the value of NNA *decreased* from 0.122 (in the second column) to 0.105 (in the third column), the value of CORTICO *decreased* from 129 (in the second column) to 11.3 (in the third column) and the value of ACTH *decreased* from 89 (in the second column) to close to zero (in the third column). We can summarise this difference as below,

Cause: Dex is increased
Effects: NNA is decreased, CORTICO is decreased, ACTH is decreased

The above effects represent a tonic action of dexamethasone on the brain to influence hypothalmic NNA and CRF release. The reduced hypothalmic NNA could contribute to the lower levels of ACTH and CORTICO, through reduced CRF. The only constraint imposed while deriving these explanations is that, a node cannot be assumed to have two different states in explaining one difference. For example, in the above difference, we need to explain three

effects. We cannot assume CRF increasing in explaining one effect and CRF decreasing in explaining another effect.

Similarly, after inducing cold swim stress on rats with dexamethasone pretreatment, the value of NNA *increased* from 0.105 (in the third column) to 0.246 (in the fifth column) and the value of CORTICO *increased* from 11.3 to 32.8. The values of ACTH were undetectable in both the cases. This means, ACTH could have increased or decreased below sensitivity level. As we are not sure, we do not need to explain the change in ACTH. These results represent a stimulatory effect of stress on the hypothalmic NNA. Increased secretion of CORTICO can be attributed to increased NNA, through increased CRF and ACTH. If we assume that ACTH increased below sensitivity levels, the hypothesis can explain the above changes. Thus the hypothesis presented in the paper [5] was able to explain both the crucial differences that were of interest to the author. However, the hypothesis is not able to explain the following two differences:

1. The difference between Experiment-1 and Experiment-4:
   Causes: Dex is increased, ColdSwimStress is increased
   Effects: NNA is increased, CORTICO is decreased, ACTH is decreased

2. The difference between Experiment-3 and Experiment-4:
   Causes: Dex is increased
   Effects: NNA is increased, CORTICO is decreased, ACTH is decreased

That is, we cannot trace back the corresponding effects to one of the causes using the proposed hypothesis (see Fig. 1). For example, for the first difference above, we can explain increase in NNA because of Cold Swim stress, but we cannot explain the other two effects: CORTICO decreasing and ACTH decreasing, as the model indicates that increase in NNA should increase ACTH.

The author overlooked the above two differences because they were not relevant to his claim in that paper. The author considered two crucial differences while constructing his hypothesis and did not include the other differences in his context. As a result of this the hypothesis presented in the paper was not able to explain some of the differences. It should be noted that the paper [5] was a refereed paper published in a prestigious international journal. Neuroendocrinology is a particularly challenging domain and the complex hypotheses discussed in the neuroendocrine papers are far from certain. They may be well accepted but are not necessarily well proved (tested against many experimental results). Researchers are always trying to present experimental data to support their own hypotheses or deny alternative hypotheses.

2.4 Incompleteness Checking

[1] describes a system, called JUSTIN (**JUST**ification **IN** context), that can be used to build and test models in neuroendocrinology. The aim of their research was to use qualitative reasoning to build and test hypotheses in the area of

neuroendocrinology. In [1], JUSTIN was used to test the model (hypothesis) of brain influence on glucose homeostatis published in [6].The paper [6] summarises six other neuroendocrine papers covering a range of hypotheses regarding brain control of glucose. The part of the summarised model (hypothesis) published in [6] is shown in Fig. 2. JUSTIN generated all possible differences and tested the model against them. It reported that 32% of the differences (between experimental treatments) could not be explained by the model. Thus JUSTIN can test the incompleteness of the current model but it cannot revise the current model such that it can explain all the observations. JustAid, described in this paper, is directed towards automating the model-revision process in the area of neuroendocrinology.

3 JustAid

JustAid allows an expert to easily build partial models of the domain, without any need of a knowledge engineer. JustAid can carry out incompleteness checking using the available observations, and if the current model cannot explain all the observations (situations), techniques from Machine Learning are used to invent possible new models that can explain all the observations. The learning algorithm can effectively use partial models of the domain while inventing new possible models. The learning algorithm also allows an expert to provide domain dependent criteria to guide the search process while looking for a new suitable model. Thus, JustAid uses domain knowledge in two different ways: initially to generate partial models and later to guide the search process while looking for a new suitable model.

In JustAid, an expert only deals with directed causal models throughout the process of model creation and modification. The graphical models are automatically converted into Horn-clause logic and a model-revision process is accomplished by a logic-based learner. As shwon in Fig. 3, the aim is to provide an intuitive and effective user interface that allows an expert to focus on the issues related to the domain and not worry about modeling constructs or underlying logical representation.

4 Model Completion Using JustAid

Space restrictions prevent a presentation of the theoretical basis for model completion in JustAid. We refer the reader to [2] and [3]. Here we simply note that this model completion is derived within a logical setting that forms the basis of Inductive Logic Programming. JustAid incorporates a new incremental learning technique that can learn definite clause logic programs from observations that are not in the form of definite clauses. In this section we provide informal description of the learning technique used in JustAid.

If the current theory cannot explain a given observation, we want to invent a new hypothesis such that the current theory along with this new hypothesis can explain an unexplained observation. Abduction uses a general rule and the

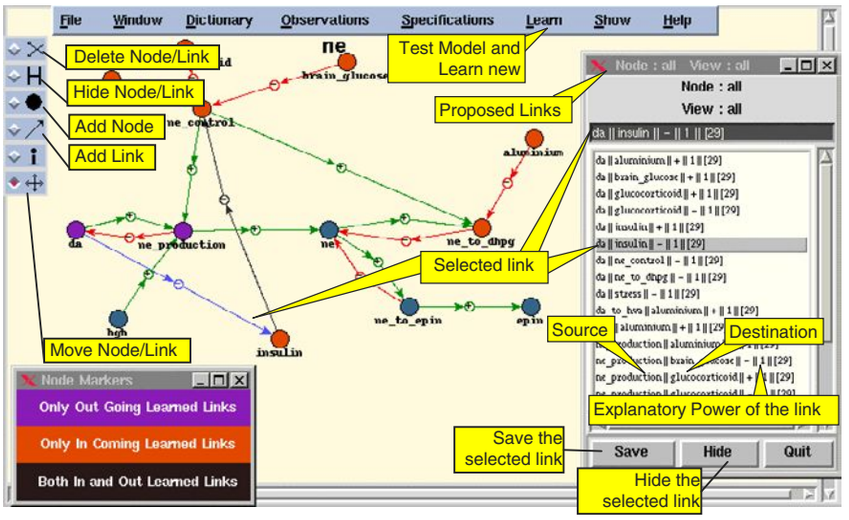


Fig. 3. Some of the features provided by the user-interface of JustAid

known conclusion to infer a specific fact that might be the cause of the known conclusion. That means, given a general rule (the current theory), we can use abduction to infer a specific fact (an abducible) that might be the cause of the known conclusion (unexplained effects). In other words, given a general rule (the current theory), we can explain the known conclusion (effects) if we can explain (derive) any one of the following:

- the known conclusion itself, or
- a possible abducible

Deduction can be used to infer possible facts given a general rule (the current theory) and a specific known fact (cause). All such inferred facts (call them deducibles) are true and we can use them while constructing a new hypothesis.

The aim now is to use these deducibles and abducibles, and construct a new hypothesis that can explain the effects or an abducible. We also want to represent this new hypothesis as a causal qualitative model and therefore we want to construct a new hypothesis that can be represented as a directed causal link(s). For the simple example shown in Fig. 4, we can explain effects if we can explain any one of the following:

- (n is increasing) and (p is decreasing) *[the known fact]*
- (n is increasing) and (f is decreasing) *[abducible]*

... ...
... ...

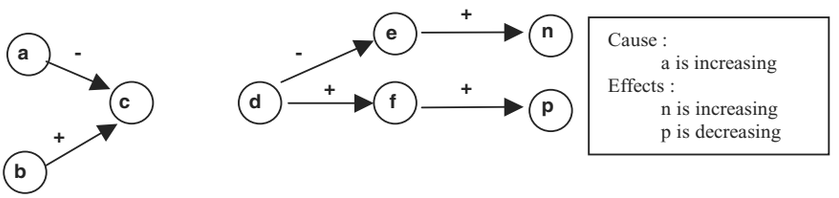


Fig. 4. A simple causal qualitative model, and an unexplained observation

... ...
... ...

- (d is decreasing) *[abducible]*

Using deduction we can derive the following fact,

- (c is decreasing)

We can now construct a new hypothesis (directed causal link) as shown in Fig. 5. The hypothesis can explain the abducible (d is decreasing), given the deduced (c is decreasing).

Fig. 5. New hypothesis

If we add the causal link shown in Fig. 5 to the current theory, we can explain the required effects, given the corresponding cause. Alternatively, we can also add other possible links that can explain some other possible abducible. However, we may want to select the link shown in Fig. 5 because it will require minimum change to the current theory (we need to add only one link).

4.1 Learning from Multiple Observations

The learning program should be able to find such common links that can fully or partially explain more than one observation (if possible). Here, *partially* means a link may explain some of the effects of an observation (difference), but not all the effects of that observation. We can use the following two indicators to find common links that can fully or partially explain more than one observation (if possible).

We say an **ExplanatoryPower$_{\textbf{effect}}$** of a link represents the number of effects that can be explained by adding that link to the current model. We can use this criterion and calculate ExplanatoryPower$_{\text{effect}}$ for every link we can induce for a given observation.

Let's assume we have n unexplained observations. Let us assume set S_i contains all the new learned links for observation-i. Let us assume set S_i also stores the maximum ExplanatoryPower$_{\text{effect}}$ for every link in set S_i. For example, let's assume there are two possible explanations for observation-i, and we need to induce the direct link from x to y for both these explanations for observation-i. Let us say, in the first explanation the ExplanatoryPower$_{\text{effect}}$ of the direct link from x to y is 1 (it can explain one effect) and in the second explanation it is 3 (it can explain 3 effects). The set S_i needs to store 3 as the ExplanatoryPower$_{\text{effect}}$ for the direct link from x to y.

We can now merge all S_i's and generate a set S_{all} that contains all the learned links for all the observations. We can calculate ExplanatoryPower$_{\text{effect}}$ of a link in the set S_{all} by adding values of ExplanatoryPower$_{\text{effect}}$ of that link in all S_i's. Thus, the value of ExplanatoryPower$_{\text{effect}}$ in the set S_{all} indicates the number of effects that can be explained by a given link in explaining all the observations.

Similarly, we can also introduce another indicator called **ExplanatoryPower$_{\textbf{observation}}$**. The aim here is to count number of observations that can be fully or partially explained by a given link. ExplanatoryPower$_{\text{observation}}$ of a link in the set S_{all} is equal to the number of S_i's that contains that link. We can now present the links in the set S_{all} to an expert by ordering them in non-increasing (descending) order on the values of ExplanatoryPower$_{\text{effect}}$ or ExplanatoryPower$_{\text{observation}}$. If the links are ordered on ExplanatoryPower$_{\text{observation}}$, the output list will have all the common links (if any) at the top of the list. If the links are ordered on ExplanatoryPower$_{\text{effect}}$, the top of the output list will contain links that can explain many effects across all the observations. Depending on the domain knowledge, an expert can select a suitable link(s) from this ordered list and modify the current model. Thus, JustAid can use multiple observations along with the current theory and propose alterations that may require minimum changes (not necessarily optimal) to the current model.

4.2 Additional Biases in JustAid

JustAid allows an expert to specify explicit biases depending on the type of model being reasoned about and the particular experiment involved. The learning algorithm uses these explicit biases in reducing the search space while looking for a suitable hypothesis. After consultations with the domain expert in the area of neuroendocrinology, the following explicit biases are implemented in the JustAid learning system.

Focus - an expert can provide a sub-graph(s) in which they are interested. The source nodes and destination nodes of new learned links should be part of this sub-graph(s).

Exclude Sub Models - an expert can provide a sub-graph(s) that should not be changed. The source nodes and destination nodes of the new learned links should not be part of this sub-graph(s).

Only Incoming/Outgoing nodes - an expert can specify node (nodes) for which the incoming or outgoing links are not possible.

Impossible Links - an expert can specify link (or links) which are not possible.

5 Experimental Results

In this section, we will examine the effectiveness of JustAid to cope with increasingly incomplete models. Given the model in Fig. 2, we are in a position to introduce artificial "model incompleteness" by random removal of links. Such removals may result in a number of observations being unexplained. Note that, even after removing a link(s), we might be still able to explain a given observation provided we could find an alternative explanation path in the reduced model. That means, it does not follow that a large number of missing links would necessarily result in a large number of unexplained observations. This is particularly true if we have many redundant links in the model. As noted by [4], the model described in Fig. 2 does contain redundant links. Note that, a domain expert may want to keep these redundant links if such relationships do exist. This point notwithstanding, in the following section we will discuss the model construction ability of JustAid for different number of unexplained observations.

5.1 Experiment Set Up

The model in [6] and Fig. 2 was constructed in JustAid. We then tested the completeness of the model against observations (differences). We take this model and the observations that it can explain. The aim after this is to remove links (randomly) from the model so that it is not possible to explain all the observations that were previously explained by the original model. This will provide an incomplete model, the corresponding unexplained observations (differences), and a set of deleted links. JustAid can then take an incomplete model and unexplained observations as input and propose possible new links.

JustAid expects experts' input in selecting links. JustAid may propose many possible new links for a given incomplete model. However, some of these links may not be suitable (or feasible) if one considers the underlying biological system. If we select a new link(s) only based on its ability to explain unexplained observations, we may construct a new model that is useless from the point of view of neuroendocrine domain, so the involvement of an expert is critical. The aim is to select previously deleted links from the suggested links. The rationale is that the expert had included these links in the original model and therefore we can consider them as suitable links from the point of view of neuroendocrine domain. Note that the aim in this research is not to replace an expert with an automated learner but to support them. This is because there may not be enough information in the model and data for the learner to propose new models that

are appropriate for the domain. In contrast, an expert knows the domain and can use additional domain knowledge (not available to the learner) in selecting a suitable new model.

In the first experiment, we removed (randomly) 11 links from the model so that it cannot explain all the observations (differences). We created 1000 such different models in which we randomly removed 11 links. For each of these models and corresponding unexplained observations (differences), we used JustAid to invent new links. The overall framework for these experiments is described below.

1. Randomly delete 11 links from the original model.
2. Using JustAid, find the number of unexplained observations (differences) and generate a list of new learned links (ordered on $\text{ExplanatoryPower}_{\text{observation}}$) that can be added to fully or partially explain one or more unexplained observations.
3. If the list contains one of the deleted links, that link is added to the model. If there is more than one deleted link in the list, the deleted link with the highest explanatory power is selected and added to the model. If there are no deleted links in the list, exit the experiment.
4. Using JustAid, find the number of unexplained observations (differences) after adding the above learned link, and calculate the number of unexplained observations that can be explained by adding the above learned link.

To compare the performance of JustAid against a random selection of a link from the deleted links, we modified the 2^{nd} and 3^{rd} steps in the above setup and repeated the above experiments by randomly selecting a deleted (suitable) link. Similarly, we repeated the above experiments by selecting a link from the highest ranked links (highest $\text{ExplanatoryPower}_{\text{observation}}$) by JustAid, irrespective of whether or not it was previously deleted.

5.2 Results

For models where 11 links are removed (randomly) from the original model, Fig. 6 shows the average number of unexplained observations explained by adding one learned link selected from the suitable (deleted) links using JustAid's suggestions (ranking). It also shows the average number of unexplained observations explained by adding one link selected randomly from the suitable (deleted) links. The error bars in the figure represents the standard error of the mean (S.E.M)[2] for each number of unexplained observations.

Note that, for the above two criteria, we select a link that was a part of the original model that explained all the observations. Thus, it is natural to expect that the selected link would explain some unexplained observations. However, as shown in Fig. 6, a link selected from the suitable (deleted) links using JustAid's suggestions could explain many more unexplained observations (differences) than a link randomly selected from the suitable (deleted) links.

[2] S.E.M = (standard deviation) / Sqrt(sample size)

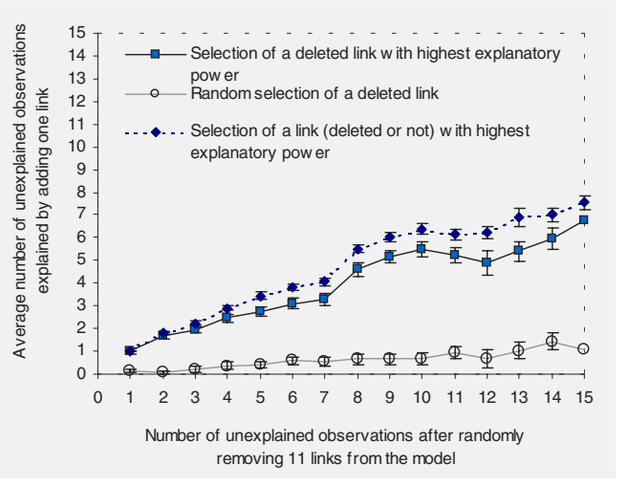


Fig. 6. Average number of unexplained observations explained by adding one link

Fig. 6 also shows that as the number of unexplained observation increases, the number of these unexplained observations explained by adding a single link, selected from the suitable links using JustAid's suggestions, also increases. JustAid tries to find a common link(s) that can explain as many observations as possible. That means, if the number of unexplained observations increases, the chances of finding a common link that can explain more unexplained observations also increases.

If we select a link only based on JustAid's ranking (irrespective of whether it was deleted or not), we can explain more unexplained observations than the other two criteria.

Note that, we used the same set of random numbers for all the three experiments described above. For example, after randomly removing 11 links, we carried out all the three experiments for the reduced model.

While the above results confirms our expectations about a single link proposed by JustAid, interest clearly remains in the complete model (that is one that can explain all the observations). The overall framework for these experiments is described below:

1. Randomly delete X (where $X \in \{5, 15\}$) links from the original model.
2. Using JustAid, generate a list of new learned links (ordered on $\text{ExplanatoryPower}_{\text{observation}}$) that can be added to fully or partially explain one or more unexplained observations.
3. If the list contains one of the deleted links, that link is added to the model. If there is more than one deleted link in the list, the deleted link with the highest explanatory power is selected and added to the model. If there are no deleted links in the list, exit the experiment.

4. Repeat steps 2 and 3 until the model explains all the observations.

Again, to compare the performance of JustAid against a random selection of a link from the deleted links, we modified the 2^{nd} and 3^{rd} steps in the above setup and repeated the above experiments. Similarly, we repeated the above experiments by selecting a link from the highest ranked links (highest $\text{Explanatory-Power}_{\text{observation}}$) by JustAid, irrespective of whether or not it was previously deleted. The results are shown in Table 2.

Table 2. Average number of new links added to explain all the observations, for different model sizes

	Average number of new links added to explain all the observations		
No of links Removed from the model	Random selection of a deleted link	Selection of a deleted link with highest explanatory power	Selection of a link (deleted or not) with highest explanatory power
5	3.99 ± 0.12	1.8 ± 0.081	1.63 ± 0.08
15	12.78 ± 0.23	4.6 ± 0.197	4.02 ± 0.158

Table 2 shows that the model does have redundant links and justAid can be used to remove them (if required). It should be noted that indeterminacy in qualitative reasoning would produce many possible explanations for a given set of effects. Therefore it might be possible to remove few links and still explain all the effects. However, if an expert is satisfied with all the links in the model, they may not choose to remove any links. It should be noted that although links may not be required for this particular set of observations, the model will generally reflect a range of other well established views about the system (world). The redundancy in biological control systems may also contribute to some of these redundant links. Thus, the issue of removing redundant links only arises when an expert does not strongly believe in the entire hypothesis and may want to explore few alternative hypotheses.

We have also empirically evaluated JustAid when used by the domain expert, to revise a real world neuroendocrine model described in [6]. The domain expert found the interaction with JustAid very friendly and stimulating. We will report the results of this study in our future publications.

6 Conclusions

We have described a new framework and a tool, called JustAid, that can assist researchers in checking that their scientific models can explain available data and that can make useful suggestions to researchers about how to improve their models. Even with the very simple causal reasoning tools described here applied to real test cases, it has been possible to point out to a researcher problems

with their models and make useful suggestions as to how they can be improved. This has been possible because a researcher is normally focussed on a specific hypothesis, whereas a computer program will search through all the material available. The experience of interacting with such a system for the researcher is not so much as interacting with a school master correcting mistakes, but interacting with a lateral thinker suggesting other things that need to be taken into account. The researcher sees the program not as pointing out errors but in checking ramifications of the model that they would not normally consider. The researcher found this interaction stimulating. If this sort of result can be found with the simple prototype we have described here, we anticipate that as these sort of tools become more developed they will have a central place in the experimental researcher's toolkit.

References

1. Feldman, B.Z., Compton, P.J., Smythe, G.A.: Hypothesis testing: an appropriate task for knowledge based systems. Proc of the 4^{th} Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada: SRDG Publications (1989)
2. Mahidadia, A., Compton, P., Sammut, C.: Helping Researchers To Construct Scientific Models - A Tool from Inductive Logic Programming. Proc of the 4^{th} Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop, Tokyo, Japan (1994)
3. Mahidadia, A.: Helping Researchers to Construct Scientific Models. PhD Thesis, School of Computer Science and Engineering, University of New South Wales, Sydney, Australia (2001)
4. Menzies, T.: Principles for Generalised Testing of Knowledge Bases. PhD Thesis, School of Computer Science and Engineering, University of New South Wales, Sydney, Australia (1995)
5. Smythe, G.A.: Hypothalamic noradrenergic activation of stress-induced Adrenocorticotropin (ACTH) release: Effects of acute and chronic dexamethasone pretreatment in rat. Experimental Clinical Endocrinology (life Sci Adv) (1987)
6. Smythe, G.A.: Brain-hypothalamus, pituitary and the endocrine pancreas. In: Samols, E. (eds.): The Endocrine Pancreas, Raven Press, New York (1989)

A General Theory of Deduction, Induction, and Learning*

Eric Martin[1], Arun Sharma[1], and Frank Stephan[2]

[1] School of Computer Science and Engineering, The University of New South Wales, Sydney, NSW 2052, Australia,
{emartin, arun}@cse.unsw.edu.au
[2] Universität Heidelberg, 69121 Heidelberg, Germany,
fstephan@math.uni-heidelberg.de

Abstract. Deduction, induction, learning, are various aspects of a more general scientific activity: the discovery of truth. We propose to embed them in a common, logical framework. First, we define a generalized notion of "logical consequence." Alternating compact and "weakly compact" consequences, we stratify the set of generalized logical consequences of a given theory in a hierarchy. Classical first-order logic is a particular case of this framework; the fact that it is all about deduction is due to the compactness theorem, and this is reflected by the collapsing of the corresponding hierarchy to the first level. Classical learning paradigms in the inductive inference literature provide other particular cases. Finite learning corresponds exactly to the first level (or level Σ_1) of the hierarchy, whereas learning in the limit corresponds to another level (namely Σ_2). More generally, strong and natural connections exist between our hierarchy of generalized logical consequences, the Borel hierarchy, and the hierarchy which measures the complexity of a formula in terms of alternations of quantifiers. It is hoped that this framework provides the foundation of a unified logic of deduction and induction, and highlights the inductive nature of learning. An essential motivation for our work is to apply the theory presented here to the design of "Inductive Prolog", a system with both deductive and inductive capabilities, based on a natural extension of the resolution principle.

1 Introduction

Let us first make a few remarks about the nature of deduction and the nature of induction, before we turn to the nature of learning. If a formula φ is a deductive consequence of a set of formulas T, it is clear to anyone that φ is a logical consequence of T, in the sense that φ is true in every model of T. Many would also agree that we can substitute "inductive" for "deductive" in the previous sentence. What is then the difference between φ being a deductive and φ being

* Eric Martin is supported by the Australian Research Council Grant A49803051. Frank Stephan is supported by the Deutsche Forschungsgemeinschaft (DFG) Heisenberg Grant Ste 967/1-1.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 228–242, 2001.

an inductive consequence of T? Well, it should be possible to discover with certainty that φ is a deductive consequence of T, if this is indeed the case. Whereas it should not be possible to discover with certainty that φ is an inductive consequence of T, if φ is not in fact a deductive consequence of T. How can we discover with certainty that φ is true on the basis of T? A natural answer is: if and only if φ is actually a logical consequence of a finite subset of T. In other words, if and only if φ is a compact logical consequence of T. On the other hand, if φ is an inductive, but not a deductive, consequence of T, then we need an infinite part of T, if not the whole of T, in order to be able to establish this fact. At this point, two questions emerge:

1. What should count as a model of T? If it is any structure (as defined in classical logic) in which every member of T is true, and if T consists of first-order formulas, then the compactness theorem shows that every logical consequence of T is actually a deductive consequence of T, and there is no scope for a proper notion of induction. Hence, we should be able to consider not all structures, but some of them. This would result in a generalized notion of logical consequence that might not be compact. Are there natural candidates for such sets of structures?
2. Suppose that the class of models of T that have been retained is such that some generalized logical consequence φ of T is not a deductive (compact) consequence of T. Is then φ automatically "promoted" to the status of inductive consequence of T? The fact that every model of T is a model of φ involves infinitely many members of T. But how difficult is it to conclude that φ is true on the basis of T? If we can define difficulty levels, should one of them be considered as "the inductive level?"

Let us now consider learning (for more details on the notions mentioned below, see [7]). We claim that the classical paradigms in the inductive inference literature are also about discovering the truth. Suppose that the underlying logical vocabulary consists of a unary predicate symbol P, together with a constant $\overline{n}$ for each natural number n. A language L can be identified with $T = \{P\overline{n} \,|\, n \in L\}$, and its complement with $\overline{T} = \{\neg P\overline{n} \,|\, n \in \mathbb{N} \setminus L\}$. A text (respect. informant) for L can then be identified with an enumeration of T (respect. $T \cup \overline{T}$). The task of discovering an r.e. index for (respect. the characteristic function of) L can be identified with the task of discovering the infinitary formula $\bigwedge T$ (respect. $\bigwedge T \wedge \bigwedge \overline{T}$). Clearly $\bigwedge T$ is a logical consequence of T—in the classical sense, hence also in any more general sense. Retaining only the structures that correspond to languages *i.e.*, the intended possible realities, will make $\bigwedge T \wedge \bigwedge \overline{T}$ a generalized consequence of T. So in both cases, identification in the limit is about discovering a particular generalized logical consequence of T, namely a formula that can be viewed upon as a description of the language to be learned. On the other hand, the task of discovering the truth of an arbitrary formula φ from a background theory is equivalent to partial classification (see [4]): a formula φ represents the class $\mathcal{C}$ of all theories T which logically imply φ in a general sense, and the partial classifier has to find out, on the basis of data from background

theory T, that φ is a generalized logical consequence of T, whenever this is true. Note the following:

1. Considering the infinite formula $P\overline{n_0} \wedge P\overline{n_1} \wedge P\overline{n_2} \wedge \ldots$ rather than the index of a Turing machine which generates $n_0, n_1, n_2 \ldots$ provides a logical representation equivalent to a representation in terms r.e. indexes. But infinite formulas are not only a technical way of embedding learning paradigms into a logical framework. It turns out that the extension of first-order languages to countable fragments of $\mathcal{L}_{\omega_1\omega}$ (see below) are the natural logical languages of our framework.
2. When learning from positive data only, there is an implicit assumption that *all* positive data are enumerated in a text for a language L. This means that the models of $\{P\overline{n} \mid n \in L\}$ to be considered should not be consistent with $P\overline{n}$ for any $n \in \mathbb{N} \setminus L$. The notion of generalized logical consequence, together with the right set of structures, should be able to accommodate this kind of property.

The previous considerations go beyond epistemological concerns about the nature of induction or learning. Indeed, the aim of this work is also to investigate the foundations of induction in AI. Current work in Inductive Logic Programming (see [14]) focuses on a very specific inductive task: discovering the minimal model of a potentially infinite set of data. We take a more general view and investigate general inductive abilities. Considering both deduction and induction as particular expressions of the art of discovering the truth opens the door to a unified framework which can provide the basis of an "Inductive Prolog". If Prolog is the deductive engine of AI, giving an agent the ability to compute solutions to existential queries, Inductive Prolog should be the deductive-inductive engine of AI, giving an agent the ability to compute solutions to existential or Σ_2 queries, such as: does there exist a chemical compound which has this effect on all molecules having this and that property?

We proceed as follows. In Section 2, we introduce the necessary notation and in Section 3, we describe the components of our framework. In Section 4, we define hierarchies of generalized logical consequences by the alternating the use of compactness and weak compactness, and show some of their properties relevant to learning paradigms. In Section 5, we investigate the relationship between the hierarchies of generalized logical consequences and formula complexity. As additional evidence of their naturalness we also demonstrate links with the Borel hierarchy. Finally, in Section 6, we show how a number of classical learning paradigms can be cast into our framework.

2 Notation

A *vocabulary* is a countable set of function symbols (possibly including constants) and predicate symbols. A vocabulary can, but does not have to, contain equality. If it does not, it is said to be *equality free*. From now on, S denotes an arbitrary countable vocabulary. For some results, assumptions on S will be made. We

denote by $\mathcal{L}^S_{\omega\omega}$ the set of all first-order S-formulas, and by $\mathcal{L}^S_{\omega_1\omega}$ the extension of $\mathcal{L}^S_{\omega\omega}$ that accepts countable nonempty conjunctions and disjunctions.[1] So for all countable nonempty $T \subseteq \mathcal{L}^S_{\omega_1\omega}$, the disjunction of all members of T, written $\bigvee T$, and the conjunction of all members of T, written $\bigwedge T$, both belong to $\mathcal{L}^S_{\omega_1\omega}$. Note that the occurrence or nonoccurrence of $=$ in S determines whether $\mathcal{L}^S_{\omega\omega}$ and $\mathcal{L}^S_{\omega_1\omega}$ are languages with or without equality. A *countable fragment of* $\mathcal{L}^S_{\omega_1\omega}$ is a countable subset $\mathcal{L}$ of $\mathcal{L}^S_{\omega_1\omega}$ which contains $\mathcal{L}^S_{\omega\omega}$, is closed under subformulas, boolean operators, and quantification.[2] From now on, $\mathcal{L}$ denotes a countable fragment of $\mathcal{L}^S_{\omega_1\omega}$. It represents the language on the basis of which the core of the theory is developed. Clearly, $\mathcal{L}^S_{\omega\omega}$ is the smallest countable fragment of $\mathcal{L}^S_{\omega_1\omega}$. The members of $\mathcal{L}^S_{\omega_1\omega}$ which are in Σ_0 or Π_0 *prenex form* are the quantifier free members of $\mathcal{L}^S_{\omega\omega}$. Let nonnull ordinal α and $\varphi \in \mathcal{L}^S_{\omega_1\omega}$ be given. We say that φ is in Σ_α (respect. Π_α) *prenex form* just in case one of the following holds:

1. φ is in Σ_β or Π_β prenex form for some $\beta < \alpha$, or
2. φ is of the form $\exists x\psi$ (respect. $\forall x\psi$) for some $\psi \in \mathcal{L}^S_{\omega_1\omega}$ which is in Σ_α (respect. Π_α) prenex form, or
3. φ is of the form $\bigvee X$ (respect. $\bigwedge X$) for some (countable) $X \subseteq \mathcal{L}^S_{\omega_1\omega}$ all of whose members are in Σ_α (respect. Π_α) prenex form.

It is easy to verify that every member of $\mathcal{L}^S_{\omega_1\omega}$ is logically equivalent to a member of $\mathcal{L}^S_{\omega_1\omega}$ which is in Σ_α prenex form for some α. If $\varphi \in \mathcal{L}^S_{\omega_1\omega}$ is logically equivalent to a closed member of $\mathcal{L}^S_{\omega_1\omega}$ which is in Σ_α (respect. Π_α) prenex form, then we say that φ is Σ_α (respect. Π_α). Note that the classical definition of a member of $\mathcal{L}^S_{\omega\omega}$ being Σ_n (respect. Π_n) for some $n \in \mathbb{N}$ is a particular case of the former.

The $\sim$ *operator* is the function $\sim\colon \mathcal{L}^S_{\omega_1\omega} \to \mathcal{L}^S_{\omega_1\omega}$ which is defined as follows. If $\varphi \in \mathcal{L}^S_{\omega_1\omega}$ is of form $\neg\psi$ for some $\psi \in \mathcal{L}^S_{\omega_1\omega}$, then $\sim(\varphi) = \psi$; otherwise $\sim(\varphi) = \neg\varphi$. Given $\Gamma \subseteq \mathcal{L}^S_{\omega_1\omega}$ and S-structure $\mathfrak{M}$, the *Γ-diagram of* $\mathfrak{M}$, denoted $D_\Gamma(\mathfrak{M})$, is the set of all members of Γ that are true in $\mathfrak{M}$. *Terms* will refer to S-terms, *formulas* to members of $\mathcal{L}$ (not $\mathcal{L}^S_{\omega_1\omega}$), *sentences* to closed formulas, and *structures* to S-structures. A *Henkin structure* is a structure all of whose individuals interpret closed terms.[3] A *Herbrand structure* is a structure each of whose individuals interprets a unique closed term.[4] Hence Herbrand structures are Henkin. When we consider a Henkin or a Herbrand structure, or a nonempty class of Henkin or Herbrand structures, we tacitly assume that S contains at least one constant.

---

[1] Given regular cardinal κ, $\mathcal{L}^S_{\kappa\omega}$ denotes the set of all S-formulas built from atomic S-formulas using boolean operators, quantifiers, and disjunctions or conjunctions over nonempty sets of cardinality smaller than κ. See [10].

[2] For more details on this definition, see [2] or [12].

[3] Henkin structures should not be confused with *Henkin models*, which is the name often given to the *general models* defined in [6]. Our notion of Henkin structure is closer to the *canonical structures* defined in [17] for Henkin's proof of the completeness of first-order logic.

[4] Herbrand structures are close to the *Herbrand models* considered in Logic Programming. See [3] or [11].

3 Components of the Theory

We denote by $\mathcal{W}$ a class of structures, the class of *possible worlds*. Classical first-order logic would take for $\mathcal{W}$ the class of all structures. We have explained that in order to address questions such as deduction versus induction, we need to be free to choose a more restrictive class of possible worlds. The discussion about learning suggests the consideration of $\mathcal{W}$ to be the class of all Henkin structures, or the class of all Herbrand structures. Henkin and Herbrand structures are interesting in many respects, and play a prominent role in Logic Programming ([11]). Given $T \subseteq \mathcal{L}$, we denote by $\mathrm{Mod}_{\mathcal{W}}(T)$ the class of all members of $\mathcal{W}$ that are models of T.

We denote by $\mathcal{O}$ a set of sentences, that we call the class of *possible observations*. For classical first-order logic, the choice of $\mathcal{O}$ would be irrelevant. Suppose we want to cast learning paradigms into this framework. For learning from positive data only, $\mathcal{O}$ will be equal to the set of all atomic sentences; for learning from both positive and negative data, $\mathcal{O}$ will be equal to the set of all basic sentences. Other examples can also be found in the literature (see for example [9]).

We denote by $\mathcal{T}$ a set of sets of sentences, that we call class of *possible theories*. This corresponds roughly to the class of possible texts in the inductive inference literature. Classical first-order logic would take for $\mathcal{T}$ the set of all sets of closed members of $\mathcal{L}^S_{\omega\omega}$.

The quintuple $(S, \mathcal{L}, \mathcal{W}, \mathcal{O}, \mathcal{T})$ contains all we need to define the fundamental concepts of this framework. We call this quintuple the *paradigm* under investigation, and we denote it by $\mathcal{P}$.

Definition 1. *Let $T \subseteq \mathcal{L}$ and $\mathfrak{M} \in \mathcal{W}$ be given. We say that $\mathfrak{M}$ is an* $\mathcal{O}$-minimal model of T in $\mathcal{W}$ *iff $\mathfrak{M} \in \mathrm{Mod}_{\mathcal{W}}(T)$ and for all $\mathfrak{N} \in \mathrm{Mod}_{\mathcal{W}}(T)$,*

$$\{\varphi \in \mathcal{O} \mid \mathfrak{N} \models \varphi\} \not\subset \{\varphi \in \mathcal{O} \mid \mathfrak{M} \models \varphi\}.$$

The discussion above about learning should justify the previous definition. A similar notion is also encountered in AI in the form of the *closed-world assumption* defined in [16] (for an overview see [5]), and of course in Logic Programming with the *least Herbrand models* (see [3,11]). Let $T \subseteq \mathcal{L}$ be given. Then T can have exactly one $\mathcal{O}$-minimal model in $\mathcal{W}$, or none, or many. We denote by $\mathrm{Mod}^{\mathcal{O}}_{\mathcal{W}}(T)$ the class of all $\mathcal{O}$-minimal models of T in $\mathcal{W}$. Note the following:

Lemma 2. *If $\mathcal{O}$ is closed under $\sim$ then for all $T \subseteq \mathcal{L}$,* $\mathrm{Mod}^{\mathcal{O}}_{\mathcal{W}}(T) = \mathrm{Mod}_{\mathcal{W}}(T)$.

We can now generalize the notion of logical consequence:

Definition 3. *Let $T \subseteq \mathcal{L}$ and $\varphi \in \mathcal{L}$ be given. We say that φ is a* logical consequence of T in $\mathcal{W}$*, and we write $T \models_{\mathcal{W}} \varphi$, iff every member of* $\mathrm{Mod}_{\mathcal{W}}(T)$ *is a model of φ. We say that φ is an* $\mathcal{O}$-minimal logical consequence of T in $\mathcal{W}$*, and we write $T \models^{\mathcal{O}}_{\mathcal{W}} \varphi$, iff every member of* $\mathrm{Mod}^{\mathcal{O}}_{\mathcal{W}}(T)$ *is a model of φ.*

The notion of $\mathcal{O}$-minimal logical consequence in $\mathcal{W}$ is the notion of generalized logical consequence we investigate; the other just proves useful. Although we develop the theory on a very broad basis, here we consider almost exclusively two cases of paradigms, that we now define.

Definition 4. *We say that* $\mathcal{P}$ *is* standard *iff* $\mathcal{T} = \{D_{\mathcal{O}}(\mathfrak{M}) \mid \mathfrak{M} \in \mathcal{W}\}$. *If* $\mathcal{P}$ *is standard and for all* $T \in \mathcal{T}$ *and sentences* φ, *either* $T \models^{\mathcal{O}}_{\mathcal{W}} \varphi$ *or* $T \models^{\mathcal{O}}_{\mathcal{W}} \neg\varphi$, *then we say that* $\mathcal{P}$ *is* ideal.

Standard paradigms are the analogues of the classical paradigms in the inductive inference literature. When no data are missing, the latter even correspond to ideal paradigms.

4 The Hierarchies of Generalized Logical Consequences

We now define the hierarchies of generalized logical consequences that are basically the fundamental object of study of this framework.[5] First we set, for all $T \in \mathcal{T}$, $\Sigma^{\mathcal{P}}_0(T) = \Pi^{\mathcal{P}}_0(T) = T$.

Definition 5. *Let nonnull ordinal* α *and* $T \in \mathcal{T}$ *be given. Suppose that* $\Pi^{\mathcal{P}}_\beta(T)$ *has been defined for all* $\beta < \alpha$. *A sentence* φ *belongs to* $\Sigma^{\mathcal{P}}_\alpha(T)$ *iff there exists finite* $E \subseteq T$ *and finite* $H \subseteq \bigcup_{\beta<\alpha} \Pi^{\mathcal{P}}_\beta(T)$ *such that for all* $T' \in \mathcal{T}$:

$$\text{if } E \subseteq T' \text{ and } T' \models^{\mathcal{O}}_{\mathcal{W}} H \text{ then } T' \models^{\mathcal{O}}_{\mathcal{W}} \varphi.$$

Definition 6. *Let nonnull ordinal* α *and* $T \in \mathcal{T}$ *be given. Suppose that* $\Sigma^{\mathcal{P}}_\alpha(T')$ *has been defined for all* $T' \in \mathcal{T}$. *A sentence* φ *belongs to* $\Pi^{\mathcal{P}}_\alpha(T)$ *iff* $T \models^{\mathcal{O}}_{\mathcal{W}} \varphi$ *and there exists finite* $E \subseteq T$ *and finite* $H \subseteq \Sigma^{\mathcal{P}}_\alpha(T)$ *such that for all* $T' \in \mathcal{T}$:

$$\text{if } E \subseteq T',\ T' \models^{\mathcal{O}}_{\mathcal{W}} H, \text{ and } T' \not\models^{\mathcal{O}}_{\mathcal{W}} \varphi \text{ then } \neg\varphi \in \Sigma^{\mathcal{P}}_\alpha(T').$$

The description of the Σ_α level is based on a compactness property: a finite information—evidence (subset of the theory) and hypotheses (sentences that have been "discovered" before)—enables to conclude that φ is an $\mathcal{O}$-minimal logical consequence of T in $\mathcal{W}$. The description of the Π_α level is based on a property of weak compactness: a finite information—of the same kind as before—enables to conclude that φ could not belong to the set of sentences that are not $\mathcal{O}$-minimal logical consequences of T in $\mathcal{W}$, without this fact being already discovered. The compactness property enables to conclude with certainty that φ is true. The property of weak compactness enables to believe confidently in φ. Of course, the certain conclusion and the confident belief in question are not absolute, they are relative to that part of the hierarchy below the level which is currently built. Only the sentences in $\Sigma^{\mathcal{P}}_1(T)$ can be discovered to be true with absolute certainty, and only the sentences in $\Pi^{\mathcal{P}}_1(T)$ can be believed to be true with absolute confidence, as far as T is reliable, for instance, in the case of standard paradigms, as far as T really contains all possible observations true in the underlying world. Note the relationship between weak compactness and the notion of refutability defined by Popper ([15]).

There are many equivalents to Definitions 5 and 6, some of them simpler, particularly when $\mathcal{P}$ is standard or ideal. For instance, it is easy to verify the following, to be used in the sequel.

[5] Since member of $\mathcal{L}^S_{\omega_1\omega}$ can contain infinitely many free variables, and due to the use of negation, it is simpler to accept only sentences in the hierarchies.

Lemma 7. *Suppose that $\mathcal{P}$ is standard. Let $T \in \mathcal{T}$ be given. A sentence φ belongs to $\Sigma_1^{\mathcal{P}}(T)$ iff there exists finite $E \subseteq T$ such that $E \models_{\mathcal{W}} \varphi$.*

Lemma 8. *Suppose that $\mathcal{P}$ is standard. Let $\alpha \neq 0$ and T in $\mathcal{T}$ be given. A sentence φ belongs to $\Pi_\alpha^{\mathcal{P}}(T)$ iff $T \models_{\mathcal{W}}^{\mathcal{O}} \varphi$ and there is $\psi \in \Sigma_\alpha^{\mathcal{P}}(T)$ such that for all $T' \in \mathcal{T}$, if $T' \models_{\mathcal{W}}^{\mathcal{O}} \psi$ and $T' \not\models_{\mathcal{W}}^{\mathcal{O}} \varphi$, then $\neg\varphi \in \Sigma_\alpha^{\mathcal{P}}(T')$.*

Lemma 9. *Suppose that $\mathcal{P}$ is ideal. Let $\alpha > 1$ and $T \in \mathcal{T}$ be given. A sentence φ belongs to $\Sigma_\alpha^{\mathcal{P}}(T)$ iff there is $\psi \in \bigcup_{\beta<\alpha} \Pi_\beta^{\mathcal{P}}(T)$ such that $\psi \models_{\mathcal{W}} \varphi$.*

Also note the following closure property of the Π_α levels of the hierarchies:

Lemma 10. *For all $\alpha \neq 0$ and $T \in \mathcal{T}$, $\Pi_\alpha^{\mathcal{P}}(T)$ is closed under (finite) disjunction and (finite) conjunction.*

When we expand the set of possible observations, we cannot lose any generalized logical consequence on any level of the hierarchy, assuming that we are dealing with ideal paradigms. Recall that in standard paradigms, the set of possible theories is uniquely determined by the set of possible worlds and the set of possible observations, and that ideal paradigms are standard.

Proposition 11. *Suppose that $\mathcal{P}$ is ideal. Let $\mathcal{P}'$ be a standard paradigm of the form $(S, \mathcal{L}, \mathcal{W}, \mathcal{O}', \mathcal{T}')$ with $\mathcal{O}$ included in $\mathcal{O}'$. Then for all ordinals α and $\mathfrak{M} \in \mathcal{W}$, $\Sigma_\alpha^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M})) \subseteq \Sigma_\alpha^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M}))$ and $\Pi_\alpha^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M})) \subseteq \Pi_\alpha^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M}))$.*

Proof. Proof is by induction. Since $\mathcal{O} \subseteq \mathcal{O}'$, $D_{\mathcal{O}}(\mathfrak{M}) \subseteq D_{\mathcal{O}'}(\mathfrak{M})$ for all $\mathfrak{M} \in \mathcal{W}$. Let $\alpha \neq 0$ be given, and suppose that $\Pi_\beta^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M})) \subseteq \Pi_\beta^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M}))$ for all $\beta < \alpha$ and $\mathfrak{M} \in \mathcal{W}$. Definition 5 implies immediately that for all $\mathfrak{M} \in \mathcal{W}$, $\Sigma_\alpha^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M})) \subseteq \Sigma_\alpha^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M}))$. Let $\mathfrak{M} \in \mathcal{W}$ and $\varphi \in \Pi_\alpha^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$ be given. By Lemma 8, choose $\psi \in \Sigma_\alpha^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$ such that for all $T \in \mathcal{T}$ with $T \models_{\mathcal{W}}^{\mathcal{O}} \psi$ and $T \not\models_{\mathcal{W}}^{\mathcal{O}} \varphi$, $\neg\varphi \in \Sigma_\alpha^{\mathcal{P}}(T)$. Let $\mathfrak{N} \in \mathcal{W}$ with $D_{\mathcal{O}'}(\mathfrak{N}) \models_{\mathcal{W}}^{\mathcal{O}'} \psi$ and $D_{\mathcal{O}'}(\mathfrak{N}) \not\models_{\mathcal{W}}^{\mathcal{O}'} \varphi$ be given. Since $\mathcal{P}$ is ideal and $\mathcal{O} \subseteq \mathcal{O}'$, $\mathcal{P}'$ is ideal as well, and we infer that $D_{\mathcal{O}}(\mathfrak{N}) \models_{\mathcal{W}}^{\mathcal{O}} \psi$ and $D_{\mathcal{O}}(\mathfrak{N}) \not\models_{\mathcal{W}}^{\mathcal{O}} \varphi$. Hence $\neg\varphi \in \Sigma_\alpha^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{N}))$ which, we have seen, implies that $\neg\varphi \in \Sigma_\alpha^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{N}))$. Since the same part of the proof implies that $\psi \in \Sigma_\alpha^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M}))$, we conclude with Lemma 8 that $\varphi \in \Pi_\alpha^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M}))$.

When learning from just positive data versus learning from both positive and negative data, the corresponding paradigms are still more strongly related. First we need a definition.

Definition 12. *Let ordinal α and $\varphi \in \mathcal{L}$ be given. We say that φ is Σ_α in $\mathcal{P}$ just in case for all $T \in \mathcal{T}$, if $T \models_{\mathcal{W}}^{\mathcal{O}} \varphi$ then $\varphi \in \Sigma_\alpha^{\mathcal{P}}(T)$. We say that φ is Π_α in $\mathcal{P}$ just in case for all $T \in \mathcal{T}$, if $T \models_{\mathcal{W}}^{\mathcal{O}} \varphi$ then $\varphi \in \Pi_\alpha^{\mathcal{P}}(T)$.*

Trivially, all sentences which are logically equivalent to the negation of a member of $\mathcal{O}$ are Π_1 in $\mathcal{P}$. We can then apply the following to $\mathcal{O}$ equal to the set of positive data only, and $\mathcal{O}'$ equal to the set of both positive and negative data, to see how formulas that are generalized logical consequences of a given theory T can be raised from some level in the hierarchy over T defined from $\mathcal{O}'$ to some level above in the hierarchy over T defined from $\mathcal{O}$.

Proposition 13. *Let paradigm $\mathcal{P}'$ be ideal and of the form $(S, \mathcal{L}, \mathcal{W}, \mathcal{O}', \mathcal{T}')$ with $\mathcal{O}'$ closed under $\sim$. Let ordinal α be given, and suppose that all members of $\mathcal{O}'$ are Π_α in $\mathcal{P}$. For all $\mathfrak{M} \in \mathcal{W}$:*

1. $\Sigma_\beta^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M})) \subseteq \Sigma_{\alpha+\beta}^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$ *for all nonnull ordinals β;*
2. $\Pi_\beta^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M})) \subseteq \Pi_{\alpha+\beta}^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$ *for all ordinals β.*

Proof. Without loss of generality we can assume that all members of $\mathcal{O}'$ are satisfiable in $\mathcal{W}$. If $\alpha = 0$ then $\mathcal{O}' \subseteq \mathcal{O}$, and we conclude immediately with Proposition 11, so suppose $\alpha \neq 0$. Proof is by induction on β. Since all members of $\mathcal{O}'$ are Π_α in $\mathcal{P}$, $D_{\mathcal{O}'}(\mathfrak{M}) \subseteq \Pi_\alpha^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$ for all $\mathfrak{M} \in \mathcal{W}$, which proves the second inclusion in the statement of the proposition for $\beta = 0$.

Let us verify that $\mathcal{P}$ is ideal. Let $\mathfrak{M} \in \mathcal{W}$ be given. By the preceding relation $D_{\mathcal{O}}(\mathfrak{M}) \models_{\mathcal{W}}^{\mathcal{O}} D_{\mathcal{O}'}(\mathfrak{M})$, hence for all $\varphi \in \mathcal{L}$, if $D_{\mathcal{O}'}(\mathfrak{M}) \models_{\mathcal{W}} \varphi$ then $D_{\mathcal{O}}(\mathfrak{M}) \models_{\mathcal{W}}^{\mathcal{O}} \varphi$. Since $\mathcal{P}'$ is ideal, $D_{\mathcal{O}'}(\mathfrak{M}) \models_{\mathcal{W}}^{\mathcal{O}'} \varphi$ or $D_{\mathcal{O}'}(\mathfrak{M}) \models_{\mathcal{W}}^{\mathcal{O}'} \neg\varphi$ for all sentences φ. Since $\mathcal{O}'$ is closed under $\sim$, $\{\varphi \in \mathcal{L} \mid D_{\mathcal{O}'}(\mathfrak{M}) \models_{\mathcal{W}} \varphi\} = \{\varphi \in \mathcal{L} \mid D_{\mathcal{O}'}(\mathfrak{M}) \models_{\mathcal{W}}^{\mathcal{O}'} \varphi\}$ by Lemma 2. We infer immediately that $D_{\mathcal{O}}(\mathfrak{M}) \models_{\mathcal{W}}^{\mathcal{O}} \varphi$ or $D_{\mathcal{O}}(\mathfrak{M}) \models_{\mathcal{W}}^{\mathcal{O}} \neg\varphi$ for all sentences φ. Hence $\mathcal{P}$ is ideal.

Let $\beta \neq 0$ be given. Suppose that for all $\gamma < \beta$ and $\mathfrak{M} \in \mathcal{W}$, $\Pi_\gamma^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M}))$ is a subset of $\Pi_{\alpha+\gamma}^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$. Let $\mathfrak{M} \in \mathcal{W}$ and $\varphi \in \Sigma_\beta^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M}))$ be given. Assume that $\beta = 1$. By Lemma 7, there exists finite $E \subseteq D_{\mathcal{O}'}(\mathfrak{M})$ such that $E \models_{\mathcal{W}} \varphi$. Since $\mathcal{P}$ is ideal, $D_{\mathcal{O}}(\mathfrak{M}) \models_{\mathcal{W}}^{\mathcal{O}} E$. As all members of E are Π_α in $\mathcal{P}$, we infer that $E \subseteq \Pi_\alpha^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$, hence $\bigwedge E \in \Pi_\alpha^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$ by Lemma 10. This, the fact that $E \models_{\mathcal{W}} \varphi$, and Lemma 9 then imply that $\varphi \in \Sigma_{\alpha+\beta}^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$. If $\beta > 1$, if follows easily from Lemma 9 and the induction hypothesis that $\varphi \in \Sigma_{\alpha+\beta}^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$. So we have shown that for all $\mathfrak{M} \in \mathcal{W}$, $\Sigma_\beta^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M})) \subseteq \Sigma_{\alpha+\beta}^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$. Let $\mathfrak{M} \in \mathcal{W}$ and $\varphi \in \Pi_\beta^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M}))$ be given. To complete the proof we show that φ belongs to $\Pi_{\alpha+\beta}^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$. By Lemma 8, choose $\psi \in \Sigma_\beta^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{M}))$ such that for all $T \in \mathcal{T}'$ with $T \models_{\mathcal{W}}^{\mathcal{O}'} \psi$ and $T \not\models_{\mathcal{W}}^{\mathcal{O}'} \varphi$, $\neg\varphi \in \Sigma_\beta^{\mathcal{P}'}(T)$. Let $\mathfrak{N} \in \mathcal{W}$ with $D_{\mathcal{O}}(\mathfrak{N}) \models_{\mathcal{W}}^{\mathcal{O}} \psi$ and $D_{\mathcal{O}}(\mathfrak{N}) \not\models_{\mathcal{W}}^{\mathcal{O}} \varphi$ be given. Since $\mathcal{P}$ and $\mathcal{P}'$ are ideal, we infer that $D_{\mathcal{O}'}(\mathfrak{N}) \models_{\mathcal{W}}^{\mathcal{O}'} \psi$ and $D_{\mathcal{O}'}(\mathfrak{N}) \not\models_{\mathcal{W}}^{\mathcal{O}'} \varphi$. Hence $\neg\varphi \in \Sigma_\beta^{\mathcal{P}'}(D_{\mathcal{O}'}(\mathfrak{N}))$, so $\neg\varphi \in \Sigma_{\alpha+\beta}^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{N}))$ as proved above. Since the same part of the proof also shows that $\psi \in \Sigma_{\alpha+\beta}^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$, we conclude with Lemma 8 that $\varphi \in \Pi_{\alpha+\beta}^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$.

5 Connections with Other Hierarchies

In order to be able to establish relations between the hierarchies of generalized logical consequences and other hierarchies, we define still a new hierarchy, where the Σ_α's levels are better behaved:

Definition 14. *For all ordinals α, the sets of sentences $\widetilde{\Sigma}_\alpha^{\mathcal{P}}$ and $\widetilde{\Pi}_\alpha^{\mathcal{P}}$ are defined by induction on α, as follows.*

1. $\widetilde{\Sigma}_0^{\mathcal{P}} = \mathcal{O} \cup \{\sim\varphi \mid \varphi \in \mathcal{O}\}$.

2. *For all ordinals* α, $\widetilde{\Pi}^{\mathcal{P}}_{\alpha} = \{\sim \varphi \,|\, \varphi \in \widetilde{\Sigma}^{\mathcal{P}}_{\alpha}\}$.
3. *Let* $\alpha \neq 0$ *be given. A sentence* φ *belongs to* $\widetilde{\Sigma}^{\mathcal{P}}_{\alpha}$ *iff there exists* $\beta < \alpha$ *with the following property. For all* $\mathfrak{M} \in \mathcal{W}$ *with* $\mathfrak{M} \models \varphi$, *there exists finite* $D \subseteq \widetilde{\Pi}^{\mathcal{P}}_{\beta}$ *such that* $\mathfrak{M} \models D$ *and* $D \models_{\mathcal{W}} \varphi$.

Informally, we will refer to the hierarchy defined above as the *uniform* hierarchy. We will need the following properties. First note that the levels of the uniform hierarchy are ordered as expected in a hierarchy of a Borel type.

Proposition 15. *For all ordinals* α, β *if* $\alpha < \beta$ *then* $\widetilde{\Sigma}^{\mathcal{P}}_{\alpha} \cup \widetilde{\Pi}^{\mathcal{P}}_{\alpha} \subseteq \widetilde{\Sigma}^{\mathcal{P}}_{\beta} \cap \widetilde{\Pi}^{\mathcal{P}}_{\beta}$.

Proof. Trivially, $\widetilde{\Sigma}^{\mathcal{P}}_{0} = \widetilde{\Pi}^{\mathcal{P}}_{0}$, which implies immediately that for all ordinals β, $\widetilde{\Sigma}^{\mathcal{P}}_{0} \cup \widetilde{\Pi}^{\mathcal{P}}_{0} \subseteq \widetilde{\Sigma}^{\mathcal{P}}_{\beta} \cap \widetilde{\Pi}^{\mathcal{P}}_{\beta}$. Let $\alpha \neq 0$ be given. For all $\beta > \alpha$, the inclusions $\widetilde{\Sigma}^{\mathcal{P}}_{\alpha} \subseteq \widetilde{\Sigma}^{\mathcal{P}}_{\beta}$ and $\widetilde{\Pi}^{\mathcal{P}}_{\alpha} \subseteq \widetilde{\Pi}^{\mathcal{P}}_{\beta}$ are straightforward. It is easily verified that $\widetilde{\Sigma}^{\mathcal{P}}_{\alpha} \subseteq \widetilde{\Pi}^{\mathcal{P}}_{\alpha+1}$. We infer that any $\varphi \in \widetilde{\Pi}^{\mathcal{P}}_{\alpha}$ is equal to $\sim \psi$ for some $\psi \in \widetilde{\Sigma}^{\mathcal{P}}_{\alpha}$, hence $\sim \varphi \in \widetilde{\Pi}^{\mathcal{P}}_{\alpha+1}$, hence $\varphi \in \widetilde{\Sigma}^{\mathcal{P}}_{\alpha+1}$. So $\widetilde{\Pi}^{\mathcal{P}}_{\alpha} \subseteq \widetilde{\Sigma}^{\mathcal{P}}_{\alpha+1}$. The result follows.

Remember that $\mathfrak{L}$ is just a fragment of $\mathfrak{L}^{S}_{\omega_1\omega}$, hence does not contain the disjunction or conjunction of any of its countable subsets. So $\bigvee X$ and $\bigwedge X$ in the closure property below are members of $\mathfrak{L}^{S}_{\omega_1\omega}$, but not necessarily members of $\mathfrak{L}$.

Lemma 16. *Let* $\alpha \neq 0$, *sentence* φ, *and countable* $X \subseteq \mathfrak{L}$ *be given.*

1. *If* $X \subseteq \widetilde{\Sigma}^{\mathcal{P}}_{\alpha}$ *and* $\models_{\mathcal{W}} (\varphi \leftrightarrow \bigvee X)$ *then* $\varphi \in \widetilde{\Sigma}^{\mathcal{P}}_{\alpha}$.
2. *If* $X \subseteq \widetilde{\Pi}^{\mathcal{P}}_{\alpha}$ *and* $\models_{\mathcal{W}} (\varphi \leftrightarrow \bigwedge X)$ *then* $\varphi \in \widetilde{\Pi}^{\mathcal{P}}_{\alpha}$.

Adding the requirement that $\mathcal{W}$ consists exclusively of Henkin structures enables to treat existential quantifiers as countable disjunctions, and universal quantifiers as countable conjunctions.

Corollary 17. *Suppose that* $\mathcal{W}$ *is a set of Henkin structures. Let formula* φ *with free variables* $x_1, \ldots, x_n$ *be given. Denote by* X *the set of all sentences of the form* $\varphi[t_1/x_1, \ldots t_n/x_n]$ *for some closed terms* $t_1, \ldots t_n$. *Let* $\alpha \neq 0$ *be given.*

1. *If* $X \subseteq \widetilde{\Sigma}^{\mathcal{P}}_{\alpha}$ *then* $\exists x_1 \ldots \exists x_n \varphi \in \widetilde{\Sigma}^{\mathcal{P}}_{\alpha}$.
2. *If* $X \subseteq \widetilde{\Pi}^{\mathcal{P}}_{\alpha}$ *then* $\forall x_1 \ldots \forall x_n \varphi \in \widetilde{\Pi}^{\mathcal{P}}_{\alpha}$.

We characterize $\widetilde{\Sigma}^{\mathcal{P}}_{1}$ and $\widetilde{\Pi}^{\mathcal{P}}_{1}$, assuming that $\mathcal{O}$ is closed under the $\sim$ operator.

Proposition 18. *Suppose that* $\mathcal{O}$ *is closed under* $\sim$. *Let sentence* φ *be given.*

1. $\varphi \in \widetilde{\Sigma}^{\mathcal{P}}_{1}$ *if and only if* $\models_{\mathcal{W}} \varphi$, *or* $\models_{\mathcal{W}} \neg\varphi$, *or there is a nonempty set* X *of finite, nonempty subsets of* $\mathcal{O}$ *such that* $\models_{\mathcal{W}} \bigvee\{\bigwedge D \,|\, D \in X\} \leftrightarrow \varphi$.
2. $\varphi \in \widetilde{\Pi}^{\mathcal{P}}_{1}$ *if and only if* $\models_{\mathcal{W}} \varphi$, *or* $\models_{\mathcal{W}} \neg\varphi$, *or there is a nonempty set* X *of finite, nonempty subsets of* $\mathcal{O}$ *such that* $\models_{\mathcal{W}} \bigwedge\{\bigvee D \,|\, D \in X\} \leftrightarrow \varphi$.

Proof. The proof is trivial if $\models_{\mathcal{W}} \varphi$ or $\models_{\mathcal{W}} \neg\varphi$, so suppose otherwise. Assume that $\varphi \in \widetilde{\Sigma}_1^{\mathcal{P}}$. Let $\mathfrak{M} \in \mathcal{W}$ be such that $\mathfrak{M} \models \varphi$. Choose a nonempty subset $D_{\mathfrak{M}}$ of $D_{\mathcal{O}}(\mathfrak{M})$ such that $D_{\mathfrak{M}} \models_{\mathcal{W}} \varphi$. Set $X = \{D_{\mathfrak{M}} \mid \mathfrak{M} \in \mathcal{W} \text{ and } \mathfrak{M} \models \varphi\}$. Then X is nonempty, and it is easy to verify that $\models_{\mathcal{W}} \bigvee\{\bigwedge D \mid D \in X\} \leftrightarrow \varphi$. Conversely, let nonempty set X of finite, nonempty subsets of $\mathcal{O}$ be such that $\models_{\mathcal{W}} \bigvee\{\bigwedge D \mid D \in X\} \leftrightarrow \varphi$. Since $\bigwedge D \in \widetilde{\Sigma}_1^{\mathcal{P}}$ for all $D \in X$, it follows from Lemma 16 that $\varphi \in \widetilde{\Sigma}_1^{\mathcal{P}}$. We conclude that *1.* holds, and *2.* is an immediate consequence.

Then we characterize the other levels:

Proposition 19. *Let $\alpha > 1$ and sentence φ be given.*

1. *$\varphi \in \widetilde{\Sigma}_\alpha^{\mathcal{P}}$ iff there is nonempty $X \subseteq \bigcup_{\beta<\alpha}(\widetilde{\Sigma}_\beta^{\mathcal{P}} \cup \widetilde{\Pi}_\beta^{\mathcal{P}})$ such that $\models_{\mathcal{W}} \bigvee X \leftrightarrow \varphi$.*
2. *$\varphi \in \widetilde{\Pi}_\alpha^{\mathcal{P}}$ iff there is nonempty $X \subseteq \bigcup_{\beta<\alpha}(\widetilde{\Sigma}_\beta^{\mathcal{P}} \cup \widetilde{\Pi}_\beta^{\mathcal{P}})$ such that $\models_{\mathcal{W}} \bigwedge X \leftrightarrow \varphi$.*

Proof. We assume that $\not\models_{\mathcal{W}} \neg\varphi$ (otherwise, the proof is trivial). Suppose that $\varphi \in \widetilde{\Sigma}_\alpha^{\mathcal{P}}$. Let $\mathfrak{M} \in \mathcal{W}$ be such that $\mathfrak{M} \models \varphi$. By Proposition 15, and since $\widetilde{\Pi}_\beta^{\mathcal{P}}$ is trivially closed under conjunction, we can choose $\beta < \alpha$ and $\psi_{\mathfrak{M}} \in \widetilde{\Pi}_\beta^{\mathcal{P}}$ such that $\psi_{\mathfrak{M}} \models_{\mathcal{W}} \varphi$. Set $X = \{\psi_{\mathfrak{M}} \mid \mathfrak{M} \in \mathcal{W} \text{ and } \mathfrak{M} \models \varphi\}$. Then $X \neq \emptyset$, and it is easy to verify that $\models_{\mathcal{W}} \bigvee X \leftrightarrow \varphi$. Conversely, let nonempty $X \subseteq \bigcup_{\beta<\alpha}(\widetilde{\Sigma}_\beta^{\mathcal{P}} \cup \widetilde{\Pi}_\beta^{\mathcal{P}})$ be such that $\models_{\mathcal{W}} \bigvee X \leftrightarrow \varphi$. Then $X \subseteq \widetilde{\Sigma}_\alpha^{\mathcal{P}}$ by Proposition 15 again, and Lemma 16 implies that $\varphi \in \widetilde{\Sigma}_\alpha^{\mathcal{P}}$. Hence *1.* holds, and *2.* is an immediate consequence.

The uniform hierarchy is more easily investigated than the hierarchies of generalized logical consequences. But our aim is to gain more insights into the latter, by transferring properties of the former. For this to be possible, we need to establish relations between the uniform hierarchy and the hierarchies of generalized logical consequences. In one direction we have:

Proposition 20. *Suppose that $\mathcal{O}$ is closed under $\sim$ and $\mathcal{P}$ is ideal. Then for all ordinals α and $T \in \mathcal{T}$, $\widetilde{\Sigma}_\alpha^{\mathcal{P}}(T) \subseteq \Sigma_\alpha^{\mathcal{P}}(T)$ and $\widetilde{\Pi}_\alpha^{\mathcal{P}}(T) \subseteq \Pi_\alpha^{\mathcal{P}}(T)$.*

Proof. Proof is by induction. Since $\mathcal{O}$ is closed under $\sim$ and $\mathcal{P}$ is standard, $\widetilde{\Sigma}_0^{\mathcal{P}}(T) = \Sigma_0^{\mathcal{P}}(T) = \widetilde{\Pi}_0^{\mathcal{P}}(T) = \Pi_0^{\mathcal{P}}(T) = T$, for all $T \in \mathcal{T}$. Let $\alpha \neq 0$ be given. Let $T \in \mathcal{T}$ be given, and suppose that $\widetilde{\Pi}_\beta^{\mathcal{P}}(T)$ is included in $\Pi_\beta^{\mathcal{P}}(T)$ for all $\beta < \alpha$. Let $\varphi \in \widetilde{\Sigma}_\alpha^{\mathcal{P}}(T)$ be given. Choose $\beta < \alpha$ and finite $D \subseteq \widetilde{\Pi}_\beta^{\mathcal{P}}$ such that $T \models_{\mathcal{W}}^{\mathcal{O}} D$ and $D \models_{\mathcal{W}} \varphi$. By induction hypothesis, $D \subseteq \Pi_\beta^{\mathcal{P}}(T)$, which implies immediately that $\varphi \in \Sigma_\alpha^{\mathcal{P}}(T)$. Now suppose that for all $T \in \mathcal{T}$, $\widetilde{\Sigma}_\alpha^{\mathcal{P}}(T) \subseteq \Sigma_\alpha^{\mathcal{P}}(T)$. Let $T \in \mathcal{T}$ and $\varphi \in \widetilde{\Pi}_\alpha^{\mathcal{P}}(T)$ be given. If for all $T' \in \mathcal{T}$, $T' \models_{\mathcal{W}}^{\mathcal{O}} \varphi$, then trivially $\varphi \in \Pi_\alpha^{\mathcal{P}}(T)$. Suppose there exists $T' \in \mathcal{T}$ such that $T' \not\models_{\mathcal{W}}^{\mathcal{O}} \varphi$. Since $\mathcal{P}$ is ideal, $T' \models_{\mathcal{W}}^{\mathcal{O}} \neg\varphi$. Hence $\neg\varphi \in \widetilde{\Sigma}_\alpha^{\mathcal{P}}(T')$. So by inductive hypothesis, $\neg\varphi$ belongs to $\Sigma_\alpha^{\mathcal{P}}(T')$, and we conclude that $\varphi \in \Pi_\alpha^{\mathcal{P}}(T)$.

The other direction of the relationship between the uniform hierarchy and the hierarchies of generalized logical consequences is more subtle. It is given by following proposition, left without proof for lack of space.

Proposition 21. *Suppose that $\mathcal{P}$ is ideal. Let $\alpha > 1$, $T \in \mathcal{T}$, and $\varphi \in \Sigma_\alpha^{\mathcal{P}}(T)$ be given. There exists $\beta < \alpha$ and $\psi \in \widetilde{\Pi}_\beta^{\mathcal{P}}(T)$ such that $\psi \models_{\mathcal{W}} \varphi$.*

The natural connection between the hierarchies of generalized logical consequences (for paradigms of a special form) and formula complexity is given by the result below, together with Proposition 20. Recall that a basic formula is an atomic formula or the negation of an atomic formula. Also note that the assumptions of the proposition below imply that $\mathcal{P}$ is ideal.

Proposition 22. *Suppose that $\mathcal{W}$ is a set of Henkin structures, $\mathcal{O}$ is the set of all basic sentences, and $\mathcal{P}$ is standard. Let $\alpha \neq 0$ be given. Every sentence in Σ_α (respect. Π_α) prenex form belongs to $\widetilde{\Sigma}_\alpha^{\mathcal{P}}$ (respect. $\widetilde{\Pi}_\alpha^{\mathcal{P}}$).*

Proof. Proof is by double induction on ordinals. Suppose that for all nonnull $\gamma < \alpha$, every sentence in Σ_γ prenex form belongs to $\widetilde{\Sigma}_\gamma^{\mathcal{P}}$ (hence every sentence in Π_γ prenex form belongs to $\widetilde{\Pi}_\gamma^{\mathcal{P}}$). Let sentence φ and ordinal β be such that β is the height of φ and φ is in Σ_α prenex form. Suppose that for all $\gamma < \beta$, every sentence of height γ which is in Σ_α prenex form belongs to $\widetilde{\Sigma}_\alpha^{\mathcal{P}}$. We now distinguish the cases corresponding to the definition of a formula being in Σ_α prenex form. Assume that φ is in Σ_γ or Π_γ prenex form for some $\gamma < \alpha$. If γ is nonnull then φ belongs to $\widetilde{\Sigma}_\gamma^{\mathcal{P}}$ or $\widetilde{\Pi}_\gamma^{\mathcal{P}}$ by inductive hypothesis, hence to $\widetilde{\Sigma}_\alpha^{\mathcal{P}}$. Suppose that $\gamma = 0$. Then φ is a quantifier free member of $\mathcal{L}_{\omega\omega}^S$. Let $\mathfrak{M} \in \mathcal{W}$ be such that $\mathfrak{M} \models \varphi$. Let D be the set of all basic sentences ψ such that either ψ or $\sim \psi$ occurs in φ, and $\mathfrak{M} \models \psi$. By the choice of $\mathcal{O}$ and the fact that $\mathcal{P}$ is standard, it is clear that $D \subseteq D_{\mathcal{O}}(\mathfrak{M})$ and $D \models \varphi$. Hence φ belongs to $\widetilde{\Sigma}_1^{\mathcal{P}}$, hence to $\widetilde{\Sigma}_\alpha^{\mathcal{P}}$. Assume that φ is of form $\exists x \psi$ for some variable x and $\psi \in \mathcal{L}$ which is in Σ_α prenex form. Set $X = \{\psi[t/x] \,|\, t \text{ closed term}\}$. Then X consists of sentences whose heights are smaller than β and which are in Σ_α prenex form, and it follows from Corollary 17 that $\varphi \in \widetilde{\Sigma}_\alpha^{\mathcal{P}}$. Suppose that φ is of form $\bigvee X$ for some (countable) set X of formulas which are in Σ_α prenex form. Then X consists of sentences whose height is smaller than β and which are in Σ_α prenex form, and it follows from Lemma 16 that $\varphi \in \widetilde{\Sigma}_\alpha^{\mathcal{P}}$.

Using some connection with the Borel hierarchy (see [18] for definitions and properties), we can exhibit natural paradigms such that the associated uniform hierarchy does not collapse to a finite level:

Proposition 23. *There exists a finite, equality free vocabulary V and a subset T of $\mathcal{L}_{\omega\omega}^V$ with the following property. Suppose that $S = V$, $\mathcal{W}$ is the set of Herbrand models of T, $\mathcal{O}$ is the set of basic sentences, and $\mathcal{P}$ is standard. Then for all nonnull $n \in \mathbb{N}$, $\widetilde{\Pi}_n^{\mathcal{P}} \setminus \widetilde{\Sigma}_n^{\mathcal{P}}$ contains a Π_n formula in $\mathcal{L}_{\omega\omega}^S$.*

Proof. Given $\sigma \in \{0,1\}^{<\mathbb{N}}$, the basic open set of the Cantor space consisting of members $\boldsymbol{c} \in \{0,1\}^{\mathbb{N}}$ that extend σ is denoted O_σ. For all nonnull $n \in \mathbb{N}$, choose total mapping $h_n : \mathbb{N}^n \to \{0,1\}^{<\mathbb{N}}$ such that the set A_n equal to $\bigcap_{i_1 \in \mathbb{N}} \bigcup_{i_2 \in \mathbb{N}} \ldots O_{h_n(i_1, i_2, \ldots, i_n)}$ is Π_n Borel, but not Σ_n Borel, in the Cantor

space. Suppose that $S = \{\overline{0}, s, \langle\rangle, P\}$ where $\overline{0}$ is a constant, s a unary function symbol, $\langle\rangle$ a binary function symbol, and P a unary predicate symbol. Given nonnull $n \in \mathbb{N}$, we denote by $\overline{n}$ the term obtained from $\overline{0}$ by n applications of s. Given $n > 2$ and terms $t_1, t_2, \ldots, t_n$, we denote by $\langle t_1, t_2, \ldots, t_n\rangle$ the term $\langle t_1, \langle t_2, \ldots, \langle t_{n-1}, t_n\rangle \ldots\rangle\rangle$. Choose a bijective mapping f from the set of closed terms into $\mathbb{N}$. Choose a surjective mapping g from the set of closed terms into $\{0,1\}^{<\mathbb{N}}$ such that for all nonnull $n \in \mathbb{N}$ and closed terms $t_1, \ldots, t_n$, $g(\langle\overline{n}, t_1, \ldots, t_n\rangle) = h_n(f(t_1), \ldots, f(t_n))$. Given $\boldsymbol{c} \in \{0,1\}^{\mathbb{N}}$, we define $\mathfrak{M}_{\boldsymbol{c}}$ to be the unique Herbrand structure which satisfies:

$$(*) \quad \text{for every closed term } t,\ \mathfrak{M}_{\boldsymbol{c}} \models Pt \text{ iff } \boldsymbol{c} \in O_{g(t)}.$$

Suppose that $\mathcal{W} = \{\mathfrak{M}_{\boldsymbol{c}} \mid \boldsymbol{c} \in \{0,1\}^{\mathbb{N}}\}$, $\mathcal{O}$ is the set of closed basic formulas, and $\mathcal{P}$ is standard. Since g is surjective, $\mathcal{W}$ clearly consists of the Herbrand models of the set of:

1. all formulas of the form Pt, where t is a closed term with $g(t) = ()$;
2. all formulas of the form $Pt \rightarrow \neg Pt'$, where t, t' are closed terms such that $g(t) \not\sqsubseteq g(t')$ and $g(t') \not\sqsubseteq g(t)$;
3. all formulas of the form $Pt \rightarrow Pt_0 \vee Pt_1$, where t, t_0, t_1 are closed terms such that $g(t_0) = g(t) \star 0$ and $g(t_1) = g(t) \star 1$.

For all nonnull $n \in \mathbb{N}$, set $\varphi_n = \forall x_1 \exists x_2 \ldots P(\langle\overline{n}, x_1, x_2, \ldots, x_n\rangle)$. Since f is bijective, it follows easily from the definition of g that for all members $\boldsymbol{c}$ of $\{0,1\}^{<\mathbb{N}}$ and nonnull $n \in \mathbb{N}$, $\mathfrak{M}_{\boldsymbol{c}} \models \varphi_n$ iff $\boldsymbol{c}$ belongs to A_n. Let $\Phi : \mathcal{L}^S_{\omega_1\omega} \rightarrow \{0,1\}^{\mathbb{N}}$ be such that:

- for all closed terms t, $\Phi(P(t)) = O_{g(t)}$;
- for all $\varphi, \psi \in \mathcal{L}^S_{\omega_1\omega}$, $\Phi(\varphi \vee \psi) = \Phi(\varphi) \cup \Phi(\psi)$ and $\Phi(\varphi \wedge \psi) = \Phi(\varphi) \cap \Phi(\psi)$;
- for all nonempty sets X of members of $\mathcal{L}^S_{\omega_1\omega}$, $\Phi(\bigvee X) = \bigcup_{\varphi \in X} \Phi(\varphi)$ and $\Phi(\bigwedge X) = \bigcap_{\varphi \in X} \Phi(\varphi)$.

Using the definition of Φ and $(*)$ above, it is easy to show by induction that for all nonnull $n \in \mathbb{N}$, the following holds. Let φ be a member of $\widetilde{\Sigma}^{\mathcal{P}}_n$ (respect. $\widetilde{\Pi}^{\mathcal{P}}_n$). There exists a closed member φ^* of $\mathcal{L}^S_{\omega_1\omega}$ built from $\mathcal{O}$ using $\vee, \wedge, \bigvee, \bigwedge$ only, and such that:

- $\Phi(\varphi^*)$ is Σ_n (respect. Π_n) Borel in the Cantor space;
- for all $\boldsymbol{c} \in \{0,1\}^{\mathbb{N}}$, $\mathfrak{M}_{\boldsymbol{c}} \models \varphi$ iff $\boldsymbol{c} \in \Phi(\varphi^*)$.

Let nonnull $n \in \mathbb{N}$ be given. By Proposition 22, φ_n belongs to $\widetilde{\Pi}^{\mathcal{P}}_n$. Suppose for a contradiction that $\varphi_n \in \widetilde{\Sigma}^{\mathcal{P}}_n$. Then $\Phi((\varphi^{\Sigma}_n)^*)$ is Σ_n Borel in the Cantor space. Moreover, we have seen that for every $\boldsymbol{c} \in \{0,1\}^{\mathbb{N}}$, $\mathfrak{M}_{\boldsymbol{c}} \models \varphi_n$ iff $\boldsymbol{c} \in \Phi((\varphi_n)^*)$, and for every $\boldsymbol{c} \in \{0,1\}^{\mathbb{N}}$, $\mathfrak{M}_{\boldsymbol{c}} \models \varphi_n$ iff $\boldsymbol{c} \in A_n$. Hence $A_n = \Phi((\varphi_n)^*)$, which contradicts the fact that A_n it not Σ_n Borel in the Cantor space.

The result in Proposition 23 can be transferred to the hierarchies of generalized logical consequences, thanks to the following corollary to Proposition 21.

Proposition 24. *Suppose that $\mathcal{O}$ is closed under $\sim$ and $\mathcal{P}$ is ideal. Let $\alpha \neq 0$ and $\varphi \in \mathcal{L}$ be such that $\varphi \in \widetilde{\Pi}^{\mathcal{P}}_{\alpha} \setminus \widetilde{\Sigma}^{\mathcal{P}}_{\alpha}$. Then $\varphi \in \Pi^{\mathcal{P}}_{\alpha}(T) \setminus \Sigma^{\mathcal{P}}_{\alpha}(T)$ for some $T \in \mathcal{T}$.*

Proof. Since $\varphi \notin \widetilde{\Sigma}^{\mathcal{P}}_{\alpha}$, neither $\models_{\mathcal{W}} \varphi$ nor $\models_{\mathcal{W}} \neg\varphi$. Suppose for a contradiction that for all $T \in \mathcal{T}$, if $\varphi \in \Pi^{\mathcal{P}}_{\alpha}(T)$ then $\varphi \in \Sigma^{\mathcal{P}}_{\alpha}(T)$. So for all $T \in \mathcal{T}$, if $T \models_{\mathcal{W}} \varphi$ then $\varphi \in \Sigma^{\mathcal{P}}_{\alpha}(T)$. We distinguish two cases.

1st Case: $\alpha = 1$. By Lemma 7, given $T \in \mathcal{T}$ with $\varphi \in \Sigma^{\mathcal{P}}_{\alpha}(T)$, we can choose finite, nonempty $D_T \subseteq T$ such that $D_T \models_{\mathcal{W}} \varphi$. Let X be the nonempty set of sets of form D_T, for all $T \in \mathcal{T}$ with $T \models_{\mathcal{W}} \varphi$. Clearly, $\models_{\mathcal{W}} \varphi \leftrightarrow \bigvee\{\bigwedge D_T \mid D_T \in X\}$. It then follows from Proposition 18 that $\varphi \in \widetilde{\Sigma}^{\mathcal{P}}_{\alpha}$, contradiction.

2nd Case: $\alpha > 1$. By Proposition 21, given $T \in \mathcal{T}$ such that $\varphi \in \Sigma^{\mathcal{P}}_{\alpha}(T)$, we can choose $\beta < \alpha$ and $\psi_T \in \widetilde{\Pi}^{\mathcal{P}}_{\beta}(T)$ such that $\psi_T \models_{\mathcal{W}} \varphi$. Let X be the nonempty set of formulas of form ψ_T, for all $T \in \mathcal{T}$ such that $T \models_{\mathcal{W}} \varphi$. Clearly, $\models_{\mathcal{W}} \varphi \leftrightarrow \bigvee X$. It then follows from Proposition 19 that $\varphi \in \widetilde{\Sigma}^{\mathcal{P}}_{\alpha}$, contradiction.

6 Connections to Learning Theory

Many classical learning paradigms can be cast in this framework; in other words, some learning paradigms are isomorphic to some (standard) paradigms (quintuples of the form $(S, \mathcal{L}, \mathcal{W}, \mathcal{O}, \mathcal{T})$), and many learnability results are equivalent to statements involving concepts defined from the notion of generalized logical consequence. We give a flavour of the connection, in the form of a few definitions and results left without proof.

The definitions usually given in the numerical setting (see [7]) are immediately adapted to the logical one. Given $T \subseteq \mathcal{L}$, a *text for* T is a sequence of members of T with at least one occurrence of every member of T. If e is a text for T and k an integer, we denote by $e[k]$ the initial segment of e of length k. A *learner* is a partial function from the set of finite sequences of members of $\mathcal{L}$, into the power set of $\mathcal{L}$. (Only uncomputable learners will be considered here, but the following material is easily relativized to computable learners.)

Definition 25 ([4]). *Let learner f and $\varphi \in \mathcal{L}$ be given. We say that f* partially identifies *φ in $\mathcal{P}$ iff for all $T \in \mathcal{T}$ and texts e for T, the following are equivalent:*

1. $T \models^{\mathcal{O}}_{\mathcal{W}} \varphi$.
2. $\{k \in \mathbb{N} \mid \varphi \notin f(e[k])\}$ *is finite.*

If f partially identifies φ in $\mathcal{P}$ and there is no $T \in \mathcal{T}$, no text e for T, and no $k_1, k_2 \in \mathbb{N}$ with $k_1 < k_2$, $\varphi \in f(e[k_1])$, and $\varphi \notin f(e[k_2])$, then we way that f partially identifies φ in $\mathcal{P}$ without mind changes.

Being at level Σ_1 of the hierarchies of generalized logical consequences (which in our view is also equivalent to being a deductive consequence of the underlying theory) and being learnable without mind changes are basically the same notion. No assumption on the paradigm $\mathcal{P}$ is needed:

Proposition 26. *Let sentence φ be given. The following are equivalent.*

1. *φ is Σ_1 in $\mathcal{P}$.*
2. *Some learner partially classifies φ in $\mathcal{P}$ without mind changes.*

Still without assumption on $\mathcal{P}$, it can be verified that level Σ_2 of our hierarchies consist of nothing but formulas that are partially identifiable:

Proposition 27. *There exists a learner f such that for all sentences φ, if φ is Σ_2 in $\mathcal{P}$ then f partially classifies φ in $\mathcal{P}$.*

The converse of Proposition 27 holds for standard paradigms (classical learning paradigms indeed correspond to standard paradigms) when the set of $\mathcal{O}$-diagrams of members of $\mathcal{W}$ is countable, as a consequence of the assumption of the following proposition together with the fact that $\mathcal{L}$ is countable.

Proposition 28. *Set $\sim \mathcal{O} = \{\sim \psi \,|\, \psi \in \mathcal{O}\}$. Suppose that $\mathcal{P}$ is standard and $\bigwedge D_{\sim\mathcal{O}}(\mathfrak{M})$ belongs to $\mathcal{L}$ for all $\mathfrak{M} \in \mathcal{W}$. Let sentence φ be given. The following are equivalent.*

1. *φ is Σ_2 in $\mathcal{P}$.*
2. *Some learner partially classifies φ in $\mathcal{P}$.*
3. *For all $T \in \mathcal{T}$ with $T \models^{\mathcal{O}}_{\mathcal{W}} \varphi$, there exists finite $D \subseteq T$ such that for all $T' \in \mathcal{T}$, if $D \subseteq T' \subseteq T$ then $T' \models^{\mathcal{O}}_{\mathcal{W}} \varphi$.*

Identification in the limit of classes of nonempty languages also has a natural analogue here. Basically, it corresponds to the hierarchies of generalized logical consequences collapsing to level Σ_2. Note the similarity of clause *3.* with the characterization of learnability of classes of nonempty recursive languages from positive data given in [1].

Proposition 29. *Set $\widetilde{\mathcal{O}} = \mathcal{O} \cup \{\sim \psi \,|\, \psi \in \mathcal{O}\}$. Suppose that $\mathcal{P}$ is standard and $\bigwedge D_{\widetilde{\mathcal{O}}}(\mathfrak{M})$ belongs to $\mathcal{L}$ for all $\mathfrak{M} \in \mathcal{W}$. The following are equivalent.*

1. *Every sentence is Σ_2 in $\mathcal{P}$.*
2. *For all $\mathfrak{M} \in \mathcal{W}$, $\bigwedge D_{\widetilde{\mathcal{O}}}(\mathfrak{M}) \in \Pi_1^{\mathcal{P}}(D_{\mathcal{O}}(\mathfrak{M}))$.*
3. *For all $T \in \mathcal{T}$, there exists finite $D \subseteq T$ such that for all $T' \in \mathcal{T}$, if $D \subseteq T'$ then $T' \not\subset T$.*

There are connections between our framework and learning paradigms with uncountably many possible worlds (as examples of such paradigms, see [8,13]). We can get results similar to the previous ones with suitable sets of assumptions.

7 Conclusion

We still haven't answered the following question: what is an inductive consequence of a theory T? We think there are good epistemological reasons to claim that a sentence φ an inductive consequence of T (in $\mathcal{P}$, assuming that T belongs to $\mathcal{T}$) just in case φ is a member of $\Pi_1^{\mathcal{P}}(T)$. So we would identify sentences obtained from T by one application of the compactness property with deductive

consequences of T, and sentences obtained from T by one application of the weak compactness property as inductive consequences of T. But the weak compactness property needs justifications coming from many directions. The fact that it enables, together with the compactness property, to build hierarchies of generalized logical consequences that have natural connections with other classical hierarchies, provides another justification. Higher levels of the hierarchies correspond to more challenging ways of discovering the truth. With this respect, level Σ_2 deserves special attention, thanks to the connection between this level and the notion of learnability in the limit. Another part of our project, an Inductive Prolog, targets precisely these formulas. Given a possible theory T and a sentence φ of the form $\exists\overline{x}\forall\overline{y}\psi(\overline{x},\overline{y})$, where ψ is a quantifier-free formula, such that $T \models^{\mathcal{O}}_{\mathcal{W}} \varphi$, the system will try to compute a sequence of terms $\overline{t}$ such that $T \models^{\mathcal{O}}_{\mathcal{W}} \forall\overline{y}\psi(\overline{t},\overline{y})$, performing top-down searches both at level Σ_2 and at level Σ_1.

References

1. D. Angluin. *Inductive Inference of Formal Languages from Positive Data.* Information and Control, 45, 1980, p. 117–135.
2. J. Barwise. *Admissible Sets and Structures.* Perspectives in Mathematical Logic, Springer-Verlag, 1975.
3. K. Doets. *From Logic to Logic Programming.* MIT Press, 1994.
4. W. Gasarch, M. Pleszkoch, F. Stephan, and M. Velauthapillai. *Classification using information.* Annals of Mathematics and Artificial Intelligence. Selected papers from ALT 1994 and AII 1994, 23:147-168, 1998.
5. M. L. Ginsberg ed. *Readings in Nonmonotonic reasoning.* Morgan Kaufmann, 1987.
6. L. Henkin. *Completeness in the theory of types.* Journal of Symbolic Logic 15:81-91, 1950.
7. S. Jain, D. N. Osherson. J. S. Royer and A. Sharma. *Systems that learn: An Introduction to Learning Theory, Second Edition.* The MIT Press, 1999.
8. K. Kelly. The Logic of Reliable Inquiry. Oxford University Press, 1996.
9. K. Kelly and C. Glymour. *Inductive inference and theory-laden data.* Journal of Philosophical Logic, 21(4), 1992.
10. H. J. Keisler. *Fundamentals of Model Theory.* In Handbook of Mathematical Logic, Ed. J. Barwise, Springer-Verlag, 1977.
11. J. W. Lloyd. Foundations of Logic Programming, 2nd edition. Springer-Verlag, 1987.
12. M. Makkai. *Admissible Sets and Infinitary Logic.* In *Handbook of Mathematical Logic*, Ed. J. Barwise, Springer-Verlag, 1977.
13. E. Martin and D. N. Osherson. *Elements of Scientific Inquiry.* The MIT Press, 1998.
14. S. H. Nienhuys-Cheng and R. de Wolf. *Foundations of Inductive Logic Programming.* Lecture Notes in Artificial Intelligence, Vol. 1228, Springer-Verlag, 1997.
15. K. Popper. *The Logic of Scientific Discovery.* Hutchinson, London, 1959.
16. R. Reiter. *On closed world data bases.* In *Logic and Data Bases*, Ed. A. Gallaire and J. Minker. Plenum, New York, 1978.
17. J. Shoenfield. *Mathematical Logic.* Addison-Wesley, 1967.
18. S. M. Srivastava. *A Course on Borel Sets.* Graduate Texts in Mathematics 180. Springer-Verlag, 1998.

Learning Conformation Rules

Osamu Maruyama[1], Takayoshi Shoudai[2], Emiko Furuichi[3], Satoru Kuhara[4], and Satoru Miyano[5]

[1] Faculty of Mathematics, Kyushu University, Fukuoka, 812-8581, Japan, om@math.kyushu-u.ac.jp
[2] Department of Informatics, Kyushu University
[3] Fukuoka Women's Junior College
[4] Graduate School of Genetic Resources Technology, Kyushu University
[5] Human Genome Center, Institute of Medical Science, University of Tokyo

Abstract. Protein conformation problem, one of the hard and important problems, is to identify conformation rules which transform sequences to their tertiary structures, called conformations. Our aim of this work is to give a concrete theoretical foundation for graph-theoretic approach for the protein conformation problem in the framework of a probabilistic learning model. We propose the conformation problem as a learning problem from hypergraphs capturing the conformations of proteins in a loose way. We consider several classes of functions based on conformation rules, and show the PAC-learnability of them. The refutable PAC-learnability of functions is discussed, which would be helpful when a target function is not in the class of functions under consideration. We also report the conformation rules learned in our preliminary computational experiments.

1 Introduction

A protein is a chain of amino acid residues that folds into a unique *native* tertiary structure under specific conditions. Biochemical experiments show that an unfolded protein spontaneously refolds into its native structure when specific conditions are restored. This is the basis for the hypothesis that the native structure of a protein can be determined from the information contained in the amino acid sequence. Under this hypothesis, various computational methods of predicting protein conformation from sequence have been proposed.

Protein conformation is analyzed in terms of free energy, where it is assumed that the free energy of a native structure of a protein is the globally minimum, which is known as "thermodynamic hypothesis." Many computational methods based on the assumption have been extensively developed. For example, Church and Shalloway [1] developed a top-down search procedure in which conformation space is recursively dissected according to the intrinsic hierarchical structure of a landscape's effective-energy barriers, and Konig and Dandekar [4] applied genetic algorithms to this problem. Another interesting heuristic method is the

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 243–257, 2001.

hydrophobic zipper method by Dill *et al.* [2]. Based on the fact that many hydrophobic contacts are topologically local, the hydrophobic zipper method randomly generates hydrophobic contacts near enough in a sequence, which serve as constraints forcing other hydrophobic contacts to be zipped up sequentially.

Inspired by this hydrophobic zipper method, but apart from the free-energy minimization problem, we introduce a *hypergraph representation* of the tertiary structure of a protein, and a *conformation rule* which is defined as a rewriting rule of hypergraphs.

Many simple conformation models in free-energy minimization problems use lattices, which are periodic graphs in two- or three-dimensional space. The conformation of a protein turns to be a self-avoiding path in the lattice in which the nodes are labeled by the amino acids. Thus the hypergraph representation representation model is a generalization of the lattice model. The degree of a node v of a hypergraph is the number of hyperedges including v, and the rank of a hyperedge e is the number of the nodes in e. Because of spatial conditions of conformations, it would be natural to impose restrictions on both of the degrees and the ranks of a hypergraph representing a tertiary structure to be bounded by constants, which is helpful in learning conformation rules. We capture the tertiary structure of a protein as a hypergraph in a loose way, from which conformation rules are extracted.

Conformation rules are repeatedly applied to a hypergraph, where the initial hypergraph is a hypergraph representing an amino acid sequence, called a chain-hypergraph. The procedure searches for a location in the current hypergraph which is applicable to a conformation rule, from local toward global as in the hydrophobic zipper method. Thus we can say that our procedure of applying conformation rules to a sequence obeys the "local to global" folding principle, which is one of the various folding principles proposed so far. The resulting hypergraph represents the structure of the protein.

We then consider the problem of learning conformation rules from hypergraph representations of proteins. A *conformation* is defined as a function from sequences to hypergraphs. Thus the problem is to learn functions from an example, that is, a pair of a protein sequence and the corresponding hypergraph representation. The PAC-learning paradigm was extended to include functions by Natarajan and Tadepalli [9] and some results on concept learning have been extended for functions [7,8].

This paper has three contributions. One is a formulation of conformation rules by using hypergraphs, and another is a polynomial-time PAC-learning algorithm for a class which is defined by this new concept of conformation rules. The other is some results on refutable PAC-learnability of functions, which would be helpful when a target function is not in the classes of functions we consider.

We have implemented the algorithms of learning conformation rules and applying conformation rules in the Python language [13]. Preliminary computational experiments have been done with using TIM barrel proteins whose data files can be downloaded from the site of Protein Data Bank (PDB) [14]. The results of the experiments are also reported.

2 Preliminaries

A *hypergraph* $H = (V, E)$ consists of a set V of nodes and a set E of *hyperedges* each of which is a nonempty subset of V. In this paper we assume that $|e| \geq 2$ for all $e \in E$ without any notice. The *rank* of H is $r(H) = \max_{e \in E} |e|$. For a node v, the *degree of* v is $d_H(v) = |\{e \in E \mid v \in e\}|$ and the *degree of* H is $d(H) = \max_{v \in V} d_H(v)$. A *chain-hypergraph* is a hypergraph $H = (V, E)$ such that $V = \{1, 2, \ldots, n\}$ for some $n \geq 1$ and each $\{i, i+1\}$ is contained in some hyperedge in E for $1 \leq i \leq n-1$, i.e., there is $e \in E$ with $\{i, i+1\} \subseteq e$. Especially, a chain-hypergraph $H = (V, E)$ is called a *rank* k *linear chain-hypergraph* if $E = \{\{i, \ldots, i+k-1\} \mid i = 1, \ldots, n-k+1\}$. For a set E of hyperedges, we call

$$\text{simplify}(E) = E - \{e \in E \mid \text{there is } e' \text{ in } E \text{ with } e \subseteq e' \text{ and } e \neq e'\}$$

the *simplification* of E. In this paper we consider a hypergraph $H = (V, E)$ whose nodes are labeled with a mapping $\psi : V \to \Delta$, where Δ is an alphabet. It is denoted by $H = (V, E, \psi)$, and called a *hypergraph over* Δ. We confuse $H = (V, E, \psi)$ with $H = (V, E)$ without any notice. Let $H = (V, E, \psi)$ and $V' \subseteq V$. For convenience, we denote by $H(V')$ the subhypergraph $\tilde{H} = (\tilde{V}, \tilde{E}, \tilde{\psi})$ of H where

- $\tilde{E} = \bigcup_{v \in V'} \{e \in E \mid v \in e\}$,
- $\tilde{V} = \bigcup_{e \in \tilde{E}} e \cup V'$,
- $\tilde{\psi} = \psi|_{\tilde{V}}$, that is, the restriction of ψ to $\tilde{V}$.

This subsection reviews some notions and results on the PAC-learnability of a class of functions by following Natarajan [7,8]. For an alphabet Ω, the set of all strings over Ω is denoted by Ω^*. The length of a string $x \in \Omega^*$ is denoted by $|x|$. For $n \geq 1$, $\Omega^{[n]} = \{x \in \Omega^* \mid |x| \leq n\}$. Here, the alphabet Ω is assumed to be finite.

Definition 1 ([7,8]). *Let F be a class of functions from a finite set X to a finite set Y. The generalized VC-dimension of F, denoted by $D(F)$, is the maximum over the sizes $|Z|$ of subsets $Z \subseteq X$ such that there exist two functions f and g in F satisfying the following conditions:*

1. *$f(x) \neq g(x)$ for all $x \in Z$.*
2. *For all $Z_1 \subseteq Z$, there exists $h \in F$ that agrees with f on Z_1 and with g on $Z - Z_1$.*

Lemma 1 ([7,8]). *Let F be a class of functions from a finite set X to a finite set Y. Then*

$$2^{D(F)} \leq |F| \leq |X|^{D(F)} |Y|^{2 \cdot D(F)}.$$

Let $f : \Omega^* \to \Omega^*$. For integers $n_1, n_2 \geq 1$, the projection $f^{[n_1][n_2]}$ of f on $\Omega^{[n_1]} \times \Omega^{[n_2]}$ is the function $f^{[n_1][n_2]} : \Omega^{[n_1]} \to \Omega^{[n_2]}$ defined by $f^{[n_1][n_2]}(x) = f(x)$ if $f(x)$ is in $\Omega^{[n_2]}$ for all x in $\Omega^{[n_1]}$. If there is some x in $\Omega^{[n_1]}$ such that $f(x)$ is not in $\Omega^{[n_2]}$, then $f^{[n_1][n_2]}$ is undefined. For a class F of functions from Ω^* to Ω^*, we define

$$F^{[n_1][n_2]} = \{f^{[n_1][n_2]} \mid f \in F,\ f^{[n_1][n_2]} \text{ is defined}\}.$$

Definition 2 ([7,8]). *Let F be a class of functions from Ω^* to Ω^* with a representation R. An algorithm $\mathcal{A}$ is said to be a polynomial-time fitting for F in representation R if the following conditions hold:*

1. *$\mathcal{A}$ is a polynomial-time algorithm taking as input a finite subset S of $\Omega^* \times \Omega^*$.*
2. *If there exists a function in F that is consistent with S, $\mathcal{A}$ outputs a name of the function in representation R.*

We say that F is of *polynomial-dimension* if there is a polynomial $p(n_1, n_2)$ in n_1 and n_2 such that $D(F^{[n_1][n_2]}) \leq p(n_1, n_2)$. We say that F is of *polynomial-expansion* if there exists a polynomial $q(n)$ such that for all $f \in F$ and $x \in \Omega^*$, $|f(x)| \leq q(|x|)$. The following theorem will be used to prove a result in Section 5 on the PAC-learnability of conformation rules.

Theorem 1 ([7,8]). *Let F be a class of functions from Ω^* to Ω^* with a representation R. F is polynomial-time PAC-learnable in R if the following hold:*

1. *F is of polynomial-dimension.*
2. *F is of polynomial-expansion.*
3. *There exists a polynomial-time fitting for F in R.*

3 Hypergraph Representation of a Protein

Let P be the protein of a primary structure $A_1 A_2 \cdots A_n$, where A_i represents the i-th amino acid residue. Its tertiary structure is usually represented by a sequence of the positions of amino acid residues in the three dimensional space as $(p_1, A_1), (p_2, A_2), \ldots, (p_n, A_n)$, where $p_i = (x_i, y_i, z_i)$ is the position of A_i for $1 \leq i \leq n$. The distance between p_i and p_j is denoted by $|p_i - p_j|$. Let Σ be the alphabet consisting of symbols representing the amino acid residues.

Let $\mu > 0$ be a real number. For a protein P with a tertiary structure (p_1, A_1), $(p_2, A_2), \ldots, (p_n, A_n)$, let $G_P^\mu = (V, E)$ be an undirected graph defined as follows:

1. $V = \{1, 2, \ldots, n\}$.
2. For any distinct i, j in V with $|p_i - p_j| \leq \mu$, $\{i, j\}$ is in E.

We call the undirected graph $G_P^\mu = (V, E)$ the *structure graph* of P with μ-range.

For positive integers k, ω, τ and $G_P^\mu = (V, E)$, let $E_{P,\mathrm{complete}}^{\mu,k,\omega,\tau}$ be the set of the hyperedges $e \subseteq V$ satisfying the following conditions:

- $2 \leq |e| \leq k$,
- $\max e - \min e + 1 \geq \tau$, that is, a restriction on the *width* of e on the sequence $1, 2, \ldots, n$,
- $G_P^\mu[e]$ is a complete graph,

where $G_P^\mu[e]$ is the node-induced subgraph of e in G_P^μ. Let

$$E_{P,\mathrm{backbone}}^{\omega} = \{\{i, i+1, \ldots, j\} \mid j = i + \omega - 1, 1 \leq i \leq n - \omega + 1\},$$

and $\psi : V \to \Sigma$ be a mapping defined by $\psi(i) = A_i$ for $1 \leq i \leq n$. Then a hypergraph $H^{\mu,k,\omega,\tau}_{P,\Sigma,\mathrm{complete}} = (V, E', \psi)$ with

$$E' = \mathrm{simplify}(E^{\mu,k,\omega,\tau}_{P,\mathrm{complete}}) \cup E^{\omega}_{P,\mathrm{backbone}}$$

is a chain-hypergraph over Σ, which is called the *hypergraph representation of P over Σ by complete graphs with μ, k, ω and τ*.

We say that an undirected graph $G = (V, E)$ where $V = \{v_0, v_1, \ldots, v_k\}$ and $E = \{\{v_0, v_i\} \mid v_i \in V, v_i \neq v_0\}$ is a *star* graph. Let $E^{\mu,k,\omega,\tau}_{P,\mathrm{star}}$ be the set of the hyperedges $e \subseteq V$ satisfying the following conditions: $2 \leq |e| \leq k$, $\max e - \min e + 1 \geq \tau$, and $G^{\mu}_{P}[e]$ is a star graph. Then a hypergraph $H^{\mu,k,\omega,\tau}_{P,\Sigma,\mathrm{star}} = (V, E'', \psi)$ with

$$E'' = \mathrm{simplify}(E^{\mu,k,\omega,\tau}_{P,\mathrm{star}}) \cup E^{\omega}_{P,\mathrm{backbone}}$$

is a chain-hypergraph over Σ, which is called the *hypergraph representation of P over Σ by star graphs with μ, k, ω and τ*.

Instead of the explicit representation with amino acid residues, it is often used to classify the amino acid residues into several categories (e.g., [2,10,11]). In order to deal with such cases, we represent a protein in a more extended way. Namely, we consider chain-hypergraphs whose nodes are labeled with some "colors", which are not necessarily the same as the amino acid residues. Let Δ be an alphabet which consists of such "colors" labeling the nodes of hypergraphs. In this paper, we assume that the tertiary structure of a protein is represented by a chain-hypergraph over some alphabet Δ in a way mentioned above.

4 Conformation Rules

In this section, we define a conformation which transforms strings over Δ to chain-hypergraphs over Δ. We denote the set of all chain-hypergraphs over Δ by $\mathcal{H}_{\Delta}$.

Definition 3. *A conformation over Δ is a function $c : \Delta^+ \to \mathcal{H}_{\Delta}$ such that $c(x) = (V, E, \psi)$ for a string $x = x_1 \cdots x_n \in \Delta^+$ satisfies $V = \{1, \ldots, n\}$ and $\psi(i) = x_i$ for $1 \leq i \leq n$.*

We give a way of completing a conformation by introducing conformation rules over Δ, which is based on hypergraph rewriting rules defined as follows: A *hypergraph rewriting rule* over Δ is a triplet $\rho = (B, A, D)$ of a hypergraph $B = (V, E, \psi)$ over Δ and subsets A and D of 2^V. The elements of A and D are called *additional* and *removable* hyperedges, respectively. The *rank* of ρ is defined to be $\max\{r(B), \max\{|a| \mid a \in A\}\}$. The *degree* of ρ is defined to be $d(B)$.

Definition 4. *Let $\rho_1 = (B_1, A_1, D_1)$ and $\rho_2 = (B_2, A_2, D_2)$ be hypergraph rewriting rules over Δ where $B_1 = (V_1, E_1, \psi_1)$ and $B_2 = (V_2, E_2, \psi_2)$. We say that ρ_1 is isomorphic to ρ_2, denoted by $\rho_1 \approx \rho_2$, if there is a bijection $\iota : V_1 \to V_2$ such that*

1. $\psi_1(v) = \psi_2(\iota(v))$ *for all* $v \in V_1$,
2. $\iota(e_1) \in E_2$ *for all* $e_1 \in E_1$, *and* $\iota^{-1}(e_2) \in E_1$ *for all* $e_2 \in E_2$,
3. $\iota(e_1) \in A_2$ *for all* $e_1 \in A_1$, *and* $\iota^{-1}(e_2) \in A_1$ *for all* $e_2 \in A_2$,
4. $\iota(e_1) \in D_2$ *for all* $e_1 \in D_1$, *and* $\iota^{-1}(e_2) \in D_1$ *for all* $e_2 \in D_2$.

Definition 5. *Let* D_Δ *be a set of hypergraph rewriting rules over* Δ. *For positive integers* P *and* Q, *we define a* $(P \times Q)$*-conformation rule* σ *over* D_Δ *as*

$$\sigma = (\beta_1, \beta_2, \ldots, \beta_P),$$

where

$$\beta_p = (\gamma_{p,1}, \gamma_{p,2}, \ldots, \gamma_{p,Q})$$

with

$$\gamma_{p,q} \subseteq D_\Delta$$

for $1 \leq p \leq P$ *and* $1 \leq q \leq Q$. $\gamma_{p,q}$ *is called the* (p,q)*-unit of* σ, *and* β_p *is the* p*th unit-sequence of* σ. D_Δ *is the domain of* σ. *The rank of* D_Δ *is defined as* $r(D_\Delta) = \max\{r(H) \mid H \in D_\Delta\}$, *and the degree of* D_Δ *is* $d(D_\Delta) = \max\{d(H) \mid H \in D_\Delta\}$. *The rank of* σ *is* $\max\{r(\gamma_{p,q}) \mid 1 \leq p \leq P,\ 1 \leq q \leq Q\}$, *and the degree of* σ *is* $\max\{d(\gamma_{p,q}) \mid 1 \leq p \leq P,\ 1 \leq q \leq Q\}$.

In this paper, we consider a rather limited hypergraph rewriting rules defined in the following way:

Definition 6. *A bundle rule over* Δ *is a hypergraph rewriting rule* $\rho = (B, A, D)$ *over* Δ *if, for* $B = (V, E, \psi)$ *over* Δ,

1. $|A| = 1$, *say* $A = \{U\}$.
2. $|U| \geq 2$.
3. $U \notin E$.
4. *For any hyperedge* e *in* E, $e \cap U \neq \emptyset$.
5. $D = \{e \in E \mid e \subset U\}$.

For short, we denote such a bundle rule $\rho = (B, A, D)$ by (B, U).

We denote by Γ_Δ the set of all bundle rules over Δ, and, for integers $k \geq 2$ and $d \geq 1$, by $\Gamma_{k,d,\Delta}$, the set of all bundle rules over Δ such that the rank is at most k and the degree is at most d.

Remark 1. Obviously, Γ_Δ is infinite. Note that $\Gamma_{k,d,\Delta}$ is finite if Δ is finite. On the other hand, $\bigcup_{k\geq 2} \Gamma_{k,d,\Delta}$ and $\bigcup_{d\geq 1} \Gamma_{k,d,\Delta}$ are infinite.

We here describe a concrete conformation, which is a function transforming strings to hypergraphs by using conformation rules. Let $\sigma = (\beta_1, \beta_2, \ldots, \beta_P)$ be a $(P \times Q)$-conformation rule over Γ_Δ where $\beta_p = (\lambda_{p,1}, \lambda_{p,2}, \ldots, \lambda_{p,Q})$ and $\lambda_{p,q} \subseteq \Gamma_{k,d,\Delta}$ for $1 \leq p \leq P$ and $1 \leq q \leq Q$. We apply σ to a string $x = x_1 \cdots x_n$ in Δ^+. For a positive integer ω, we start with a rank ω linear chain-hypergraph $H = (V, E, \psi)$, that is, $V = \{1, \ldots, n\}$, $\psi_s(i) = x_i$ for $1 \leq i \leq n$, and $E = \{\{i, \ldots, i + \omega - 1\} \mid 1 \leq i \leq n - \omega + 1\}$. At the pth stage ($1 \leq p \leq P$), the pth unit-sequence β_p of σ is used in the following way. In each stage, a window

on the node sequence $1, 2, \ldots, n$ corresponding to the string $x = x_1 \cdots x_n$ is an interval of the sequence, and enlarged from smaller to larger. The initial window size is specified by τ. For each window size, the window is slided from left to right on V. Suppose a window of size $w(\geq \tau)$ at position i, that is, an interval $[i, \ldots, i+w-1]$ consisting of consecutive w nodes in V. Let $q = w - \tau + 1$, whose range is from 1 to Q. The bundle rules in the (p, q)-unit $\gamma_{p,q}$ of σ are applied to create new hyperedges e such that e consists of only nodes in $[i, \ldots, i+w-1]$ and i and $i + w - 1$ are in e. A new creation of a hyperedge e in the window depends on a local structure around e in the current hypergraph $H = (V, E, \psi)$. Namely, we consider a subhypergraph $H(e)$. A new hyperedge e will be created if there is a bundle rule $(B, U) \in \gamma_{p,q}$ which is isomorphic to $(H(e), e)$. After creating all new hyperedges in the process of sliding the window from left to right, these hyperedges are added to E and the proper subsets of them are deleted from E, and this window sliding process is repeated after the window is enlarged. A formal description is given in Fig. 1.

Input: a $(P \times Q)$-conformation rule $\sigma = (\beta_1, \ldots, \beta_P)$ over Γ_Δ where $\beta_p = (\gamma_{p,1}, \gamma_{p,2}, \ldots, \gamma_{p,Q})$ and $\gamma_{p,q} \subseteq \Gamma_\Delta$ for $1 \leq p \leq P$ and $1 \leq q \leq Q$, positive integers ω and τ with $\omega < \tau$, and a string $x = x_1 \cdots x_n$ in Δ^+
Output: a hypergraph $H = (V, E, \psi)$
Procedure: $\mathcal{CONFORM}(\omega, \tau, \sigma, x)$

```
let k be the rank of σ
V = {1, ..., n}
let ψ be a mapping defined by ψ(i) = x_i for 1 ≤ i ≤ n
E = {{i, ..., i + ω − 1} | 1 ≤ i ≤ n − ω + 1}
H = (V, E, ψ)     # linear chain-hypergraph of rank ω
for p from 1 to P:
    for q from 1 to min{Q, n}:
        w = τ + q − 1     # w is the window size
        A = ∅; D = ∅
        foreach i from 1 to n − w + 1:
            j = i + w − 1
            foreach e ⊆ {i, ..., j} such that i, j ∈ e and |e| ≤ k:
                if a bundle rule (H(e), e) ≈ ρ for some ρ in γ_{p,q}:
                    add e to A
                    add the proper subsets of e in E to D
        E = E ∪ A \ D
```

Fig. 1. Algorithm $\mathcal{CONFORM}$

The graph G given in Fig. 2 is an example of the graphs which cannot be generated by any $(1, Q)$-conformation rule for any Q.

The following proposition is obvious by definitions:

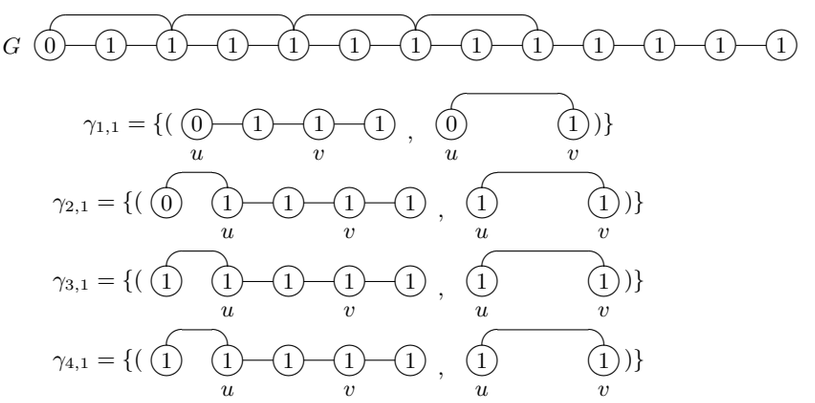


Fig. 2. $\sigma = ((\gamma_{1,1}), (\gamma_{2,1}), (\gamma_{3,1}), (\gamma_{4,1}))$ which is a (4×1)-conformation rule over $\Gamma_{2,4,\{0,1\}}$ generating the graph G

Proposition 1. *Let σ be a $(P \times Q)$-conformation rule over $\Gamma_{k,d,\Delta}$, ω and τ be positive integers with $\omega < \tau$, and $x \in \Delta^+$. The hypergraph $\mathcal{CONFORM}(\omega, \tau, \sigma, x)$ given in Fig. 1 is a chain-hypergraph over Δ of at most rank k.*

Definition 7. *For a $(P \times Q)$-conformation rule σ over Γ_Δ and positive integers ω and τ with $\omega < \tau$, we define a conformation $c_\sigma^{\omega,\tau}$ as a function from Δ^+ to the set of chain-hypergraphs over Δ, by $c_\sigma^{\omega,\tau}(x) = \mathcal{CONFORM}(\omega, \tau, \sigma, x)$ for $x \in \Delta^+$.*

5 PAC-Learning of Conformation

For a positive integer n, let $\mathcal{H}_\Delta^{[n]}$ be the set of all chain-hypergraphs over Δ with at most n nodes. By $c_\sigma^{\omega,\tau[n]}$ we denote a function $c_\sigma^{\omega,\tau[n]} : \Delta^{[n]} \to \mathcal{H}_\Delta^{[n]}$ obtained by restricting $c_\sigma^{\omega,\tau}$ to $\Delta^{[n]}$.

For integers $\omega \geq 2$, $\tau > \omega$, $P, Q \geq 1$, and an alphabet Δ, let

$$\mathcal{C}_\Delta^{\omega,\tau,P,Q} = \{c_\sigma^{\omega,\tau} \mid \sigma \text{ is a } (P \times Q)\text{-conformation rule over } \Gamma_\Delta\}.$$

As noted in Remark 1, the alphabet Γ_Δ is infinite even if Δ is finite. This makes a trouble in discussing the PAC-learnability of a class of conformations. However, if we restrict the rank and degree of conformation rules to constant integers k and d, respectively, the alphabet $\Gamma_{k,d,\Delta}$ is finite for finite alphabets Δ. Let

$$\mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,Q} = \{c_\sigma^{\omega,\tau} \mid \sigma \text{ is a } (P \times Q)\text{-conformation rule over } \Gamma_{k,d,\Delta}\}$$

for integers $k \geq 2, d \geq 1, \omega \geq 2, \tau > \omega, P \geq 1$ and $Q \geq 1$.

Our main result is the following theorem:

Theorem 2. *The class $\mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,Q}$ is polynomial-time PAC-learnable.*

Theorem 3. *The class* $\bigcup_{R\geq 1} \mathcal{C}_{k,d,\Delta}^{\omega,\tau,1,R}$ *is polynomial-time PAC-learnable.*

We can prove these theorems by showing that these classes satisfy three conditions in Theorem 1.

For an integer $k \geq 2$, a hypergraph $H = (V, E, \psi)$ of rank k with $n = |V|$ can be expressed under an appropriate encoding as a string over Δ whose length is polynomially bounded with respect to n. Thus we regard a conformation c over Δ as a function from Δ^+ to Δ^+. Therefore we can see that any class of conformations over Δ is of polynomial-expansion.

Next we show that $\mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,Q}$ and $\bigcup_{R\geq 1} \mathcal{C}_{k,d,\Delta}^{\omega,\tau,1,R}$ are of polynomial-dimension. Let

$$\mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,Q^{[n]}} = \{c_\sigma^{\omega,\tau^{[n]}} \mid \sigma \text{ is a } (P \times Q)\text{-conformation rule over } \mathit{\Gamma}_{k,d,\Delta}\}.$$

By Lemma 1, it suffices to show that $\left|\mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,Q^{[n]}}\right|$ and $\left|\bigcup_{R\geq 1} \mathcal{C}_{k,d,\Delta}^{\omega,\tau,1,R^{[n]}}\right|$ are bounded by $2^{p(n)}$ for some polynomial $p(n)$. A $(P \times Q)$-conformation rule σ over $\mathit{\Gamma}_{k,d,\Delta}$ can be considered as a $P \times Q$ matrix whose elements are subsets of $\mathit{\Gamma}_{k,d,\Delta}$. Since $|\mathit{\Gamma}_{k,d,\Delta}|$ is a finite constant, say δ, we have $\left|\mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,Q^{[n]}}\right| \leq \left(2^\delta\right)^{P\cdot Q}$, that is, $\left|\mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,Q^{[n]}}\right|$ is also bounded by a finite constant. It should be noted here that $\bigcup_{R\geq 1} \mathcal{C}_{k,d,\Delta}^{\omega,\tau,1,R^{[n]}} = \bigcup_{n\geq R\geq 1} \mathcal{C}_{k,d,\Delta}^{\omega,\tau,1,R^{[n]}}$. Thus, we can see that $\left|\bigcup_{R\geq 1} \mathcal{C}_{k,d,\Delta}^{\omega,\tau,1,R^{[n]}}\right| \leq \left(2^\delta\right)^{P\cdot n}$, which is bounded by $2^{p(n)}$ for some polynomial $p(n)$.

Finally we discuss polynomial-time fittings for $\mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,Q}$ and $\bigcup_{R\geq 1} \mathcal{C}_{k,d,\Delta}^{\omega,\tau,1,R}$. It is trivial that there is a polynomial-time fitting for $\mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,Q}$ since the cardinality of the class is a finite constant.

We then describe a polynomial-time fitting $\mathcal{B}$ for $\bigcup_{R\geq 1} \mathcal{C}_{k,d,\Delta}^{\omega,\tau,1,R}$ by employing the algorithm $\mathcal{EXTRACT}$ given in Fig. 3. Given chain-hypergraphs $H_1 = (V_1, E_1, \psi_1), \ldots, H_t = (V_t, E_t, \psi_t)$ over Δ and positive integers ω and τ with $\tau > \omega$, the algorithm $\mathcal{B}$ computes, for $1 \leq h \leq t$, a conformation rule over $\mathit{\Gamma}_\Delta$, $\hat{\sigma}^{(h)} = \mathcal{EXTRACT}(\omega, \tau, N, H_h)$, where $N = \max_{1\leq h\leq t} |V_h|$. We denote by $\hat{\gamma}_{1,q}^{(h)}$ the $(1, q)$-unit of $\hat{\sigma}^{(h)}$ for $1 \leq q \leq N$. For each q with $1 \leq q \leq N$, let $\hat{\gamma}_{1,q} = \bigcup_{1\leq h\leq t} \hat{\gamma}_{1,q}^{(h)}$, and $\hat{\sigma} = ((\hat{\gamma}_{1,1}, \hat{\gamma}_{1,2}, \ldots, \hat{\gamma}_{1,N}))$. The algorithm $\mathcal{B}$ outputs $\hat{\sigma}$ from $H_1, \ldots, H_t$. Obviously, Q runs in polynomial time since the rank of conformation rules is a constant k.

If $H_1 = \mathcal{CONFORM}(\omega, \tau, \sigma, s_1)$, $H_2 = \mathcal{CONFORM}(\omega, \tau, \sigma, s_2)$, $\ldots$, $H_t = \mathcal{CONFORM}(\omega, \tau, \sigma, s_t)$ for some $(1, Q)$-conformation rule σ over $\mathit{\Gamma}_{k,d,\Delta}$ and strings $s_1, s_2, \ldots, s_t \in \Delta^+$, then we can show that $H_h = \mathcal{CONFORM}(\omega, \tau, \hat{\sigma}, s_h)$ for $1 \leq h \leq t$, which means that $\mathcal{CONFORM}(\omega, \tau, \hat{\sigma}, \cdot)$ is consistent with the examples $\{(s_i, H_i) \mid 1 \leq i \leq t\}$. For $1 \leq h \leq t$ and $1 \leq q \leq N$, let

- $C_{h,q}$ be the contents of E just after the qth iteration of the **for**-loop on q of the 1st iteration of the **for**-loop on p of $\mathcal{CONFORM}(\omega, \tau, \sigma, s_h)$ has been finished if $q \leq \min\{Q, |s_h|\}$, $C_{h,q} = C_{h,q-1}$ otherwise.

Input: a chain-hypergraph $H = (V, E, \psi)$ over Δ of rank k, and positive integers ω, τ and R
Output: a conformation rule $\sigma = (\beta_1)$ over Γ_Δ of rank k where $\beta_1 = (\gamma_{1,1}, \gamma_{1,2}, \ldots, \gamma_{1,R})$ with $\gamma_{1,q} \subseteq \Gamma_\Delta$ for $1 \leq q \leq R$
Procedure: $\mathcal{EXTRACT}(\omega, \tau, R, H)$
  $n = |V|$
  $\tilde{E} = \{\{i, \ldots, i+\omega-1\} \mid 1 \leq i \leq n-\omega+1\}$
  $\tilde{H} = (V, \tilde{E}, \psi)$
  for q **from** 1 **to** R:
    $w = \tau + q - 1$
    $A = \emptyset$
    $D = \emptyset$
    foreach i **from** 1 **to** $n - w + 1$:
      $j = i + w - 1$
      foreach $U \subseteq \{i, \ldots, j\}$ such that $i, j \in U$ and $|U| \leq k$:
        if $U \in E$:
          $\rho = (\tilde{H}(U), U)$
          add ρ to γ_q
          add U to A
          add the proper subsets of U in $\tilde{E}$ to D
    $\tilde{E} = \tilde{E} \cup A \setminus D$

Fig. 3. Algorithm $\mathcal{EXTRACT}$

- $E_{h,q}$ be the contents of $\tilde{E}$ just after the qth iteration of the **for**-loop of $\mathcal{EXTRACT}(\omega, \tau, N, H_h)$ has been finished.
- $\hat{C}_{h,q}$ be the contents of E just after the qth iteration of the **for**-loop on q of the 1st iteration of the **for**-loop on p of $\mathcal{CONFORM}(\omega, \tau, \hat{\sigma}, s_h)$ has been finished if $q \leq |s_h|$, $C_{h,q} = C_{h,q-1}$ otherwise.

For convenience, let $C_{h,0} = E_{h,0} = \hat{C}_{h,0} = \{\{i, \ldots, i+\omega-1\} \mid 1 \leq i \leq |s_h|-\omega+1\}$ for $1 \leq h \leq t$, which the initial hyperedges in the algorithm $\mathcal{CONFORM}$ and $\mathcal{EXTRACT}$ on the string s_h. It is not hard to prove by induction on q

$$C_{h,q} = E_{h,q} = \hat{C}_{h,q}$$

for $1 \leq h \leq t$. This completes the proof.

The following theorem can be shown in a similar way and we omit its proof.

Theorem 4. *The class* $\bigcup_{R \geq 1} \mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,R}$ *is polynomial-time PAC-learnable.*

6 Refutably PAC-Learning Functions

In this section, we introduce the refutability of PAC-learning algorithms on functions. The refutability of PAC-learning algorithms on concepts have been already

discussed in [5,6]. PAC-learning algorithms having the ability to refute classes which do not seem to include a target function would be helpful in dealing with real data.

Let f be a function from Ω^* to Ω^*, F be a class of functions from Ω^* to Ω^*, and P be a probability distribution on Ω^*.

We define $opt_f(P, F)$ by

$$opt_f(P, F) = \min_{f' \in F} P(f' \triangle f).$$

We can see that if $f \in F$ then $opt_f(P, F) = 0$ for any P.

Definition 8. *Let F be a class of functions from Ω^* to Ω^*. A function class F is polynomial-sample refutably learnable if there exist an algorithm $\mathcal{A}$ and a polynomial $p(\cdot, \cdot, \cdot, \cdot)$ which satisfy the following conditions:*

1. *The algorithm $\mathcal{A}$ takes as input parameters $\varepsilon, \varepsilon', \delta \in (0, 1)$ and $n \geq 1$. We call ε' a refutation accuracy parameter.*
2. *Let f be a target function from Ω^* to Ω^* and P an arbitrary and unknown probability distribution on Ω^*. The algorithm $\mathcal{A}$ takes a sample of size $p(1/\varepsilon, 1/\varepsilon', 1/\delta, n)$ using a subroutine $EX(f, P)$, which at each call produces a single example for f according to P.*
3. *If $opt_f(P, F) = 0$ then $\mathcal{A}$ outputs a function $g \in F$ which satisfies $P(f \triangle g) < \varepsilon$ with probability at least $1-\delta$. If $opt_f(P, F) \geq \varepsilon'$ then $\mathcal{A}$ refutes the function class F with probability at least $1 - \delta$.*

Theorem 5. *If a class F of functions is of polynomial dimension, then F is polynomial-sample refutably learnable.*

By this theorem the followings hold:

Corollary 1. *The classes $\mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,Q}$ and $\bigcup_{R \geq 1} \mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,R}$ are polynomial-sample refutably learnable.*

Since F is of polynomial dimension, there exists a polynomial $poly(\cdot, \cdot)$ such that $\log_2 |F^{[n_1][n_2]}| \leq poly(n_1, n_2)$ for any $n_1, n_2 \geq 1$. We construct the algorithm described in Figure 4.

We introduce a *refutation threshold parameter* $\eta \in (0, 1)$ so that a learning algorithm produces an approximate function instead of refuting F when the minimum error $opt_f(P, F)$ is small enough.

Definition 9. *Let F be a class of functions from Ω^* to Ω^*. A function class F is polynomial-sample strongly refutably learnable if there exist an algorithm $\mathcal{A}$ and a polynomial $p(\cdot, \cdot, \cdot, \cdot)$ which satisfy the following conditions:*

1. *The algorithm $\mathcal{A}$ takes as input parameters $\varepsilon, \varepsilon', \delta, \eta \in (0, 1)$ and $n \geq 1$.*
2. *Let f be a target function from Ω^* to Ω^* and P an arbitrary and unknown probability distribution on Ω^*. The algorithm $\mathcal{A}$ takes a sample of size $p(1/\varepsilon, 1/\varepsilon', 1/\delta, n)$ using a subroutine $EX(f, P)$, which at each call produces a single example for f according to P.*

Input: $\varepsilon, \varepsilon', \delta, n_1, n_2$
Procedure:
let m = $\lceil (1/\varepsilon + 1/\varepsilon')(1/\delta + poly(n_1, n_2)) \rceil$
make m calls of EX
let S be the set of examples seen
if there is a function $g \in F$ consistent with S:
return g
else
refute F

Fig. 4. Refutable algorithm $\mathcal{A}_{RefuteBySampleComplexity}(\varepsilon, \varepsilon', \delta, n_1, n_2)$

3. *If* $opt_f(P, F) \leq \eta$ *then* $\mathcal{A}$ *outputs a concept* $g \in F$ *which satisfies* $P(f \triangle g) < \eta + \varepsilon$ *with probability at least* $1 - \delta$. *If* $opt_f(P, F) \geq \eta + \varepsilon'$ *then* $\mathcal{A}$ *refutes the function class* F *with probability at least* $1 - \delta$.

Theorem 6. *If a class* F *of functions is of polynomial dimension, then* F *is polynomial-sample strongly refutably learnable.*

Corollary 2. *The classes* $\mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,Q}$ *and* $\bigcup_{R \geq 1} \mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,R}$ *are polynomial-sample strongly refutably learnable.*

We construct the algorithm described in Figure 5. We denote by $d(g, S)$ the number of examples in S with which g does not agree.

Input: $\varepsilon, \varepsilon', \delta, \eta, n_1, n_2$
Procedure:
$\kappa = \min\{\varepsilon, \varepsilon'\}$
$m = \left\lceil 4\left(1/\varepsilon^2 + 1/\varepsilon'^2\right)(1/\delta + poly(n_1, n_2)) \right\rceil$
make m calls of EX
let S be the set of examples seen
if there is a function $g \in F$ with $d(g, S) \leq \lfloor m(\eta + (1/2)\kappa) \rfloor$ **then**
return g
else
refute F

Fig. 5. Strongly refutable algorithm $\mathcal{A}_{StronglyRefuteBySampleComplexity}(\varepsilon, \varepsilon', \delta, \eta, n_1, n_2)$

We can easily see that F is of polynomial dimension if F is polynomial-sample refutably learnable or polynomial-sample strongly refutably learnable. Therefore the following three statements are equivalent:

1. F is of polynomial dimension.
2. F is polynomial-sample refutably learnable.
3. F is polynomial-sample strongly refutably learnable.

7 Experiments

In this section, we report our preliminary computational experiments on learning conformation rules from hypergraphs representing tertially structures of proteins. We have implemented the PAC-learning algorithm shown in the algorithms $\mathcal{CONFORM}(\omega, \tau, \sigma, x)$ and $\mathcal{EXTRACT}(\omega, \tau, R, H)$ in the Python language [13].

7.1 Method of Experiments

The hypergraph representation of a protein over Δ by *star* graphs are used with μ, k, ω, τ specified as follows: $\mu = 5.8$Å, $k = 10$, $\omega = 5$ and $\tau = 8$. The choice of the alphabet Δ for labeling the nodes of a hypergraph is one of the key to experiments. The alphabet Δ represents a classification of amino acid residues. In Hart and Istrail [3], they used the hydrophobic-hydrophilic model that regards a protein as a linear chain amino acid residues that are of two types H (hydrophobic) and P (hydrophilic). However some amino acids are neither hydrophobic nor hydrophilic. In our experiments, Δ is set to $\{H, P, N\}$, where the amino acid residues are assigned as follows:

H : ALA, CYS, ILE, LEU, MET, PHE, TRP, VAL,
P : ARG, ASN, ASP, GLN, GLU, LYS, PRO, ASX, GLX,
N : GLY, HIS, SER, THR, TYR.

The class of conformations, $\mathcal{C}_{k,d,\Delta}^{\omega,\tau,P,Q}$ where $P = 1$ and $Q = 2$, is considered in the experiments (Since the degree bound d is not important rather than the rank bound k, d is unlimited.). Given examples $(s_1, H_1), \ldots, (s_t, H_t)$, the polynomial-time fitting $\mathcal{B}$, used to prove Theorem 3, outputs a $(1, 2)$-conformation rule $\hat{\sigma}$, which is applied in $\mathcal{CONFORM}(\omega, \tau, \hat{\sigma}, x)$ for a sequence x.

To evaluate how a hypergraph predicted by $\mathcal{CONFORM}$ is similar to the target hypergraph, we compare them *hyperedge by hyperedge*. To this end, we define a similarity between hyperedges as follows: Let $g \geq 0$ and $0 \leq \kappa \leq 1$, and subsets E_1 and E_2 of 2^V, where $V = \{1, 2, \ldots, n\}$. For $e_1 \in E_1$ and $e_2 \in E_2$, we say that e_1 is (g, κ)-*similar* to e_2 if $\min e_2 - g \leq \min e_1 \leq \min e_2 + g$ and $e_1 \triangle e_2 \leq \kappa$. We denote $Sim_{g,\kappa}(E_1, E_2) = |\{e_1 \in E_1 \mid e_1 \text{ is } (g, \kappa)\text{-}similar \text{ to } e_2 \in E_2\}|$

TIM-barrel proteins have high regulatory conformations, which are composed by eight parallel β-sheets forming a barrel structure [12]. We downloaded PDB files of TIM-barrel proteins from the site of PDB [14], which are screened out. The 15 proteins remains, whose tertiary structures are fully determined and composed of a single chain of amino acids.

In our experiments, the following small modification has been done: for a bundle rule $\rho = (B, A, D)$ where $A = \{U\}$, D is set to $\emptyset$ instead of $\{e \in E \mid e \subset U\}$, which affects nothing but would enable to attain more detailed conformation rules.

7.2 Evaluation

We have executed two kinds of experiments. One is self-conformation, that is, for a single protein p, a $(1, 2)$-conformation rule α is learned from the hypergraph

representation of p, and used in $\mathcal{CONFORM}$ with the sequence of p. Another is the case where a $(1,2)$-conformation rule α is extracted from 14 TIM-barrel proteins, and applied to the remaining one.

In self-conformation, the successful results are attained. Let $H_T = (V, E_T, \psi)$ and $H_P = (V, E_P, \psi)$ be a target and a predicted hypergraph, respectively. For a set S, by S^c we denote the complement of S. We give a typical results of self-conformation test in Tab. 1. Since the experiment is going well under the window sizes 7 and 8, the experiment should be continued with the window sizes over 8. However, if it is done, the procedure does not finish in a practical time. The task of hypergraph matching is repeatedly done in our procedure. An efficient and practical algorithm for the problem of hypergraph isomorphism should be developed, which would be one of the future works.

Table 1. Result of self-conformation with protein 4ALD, whose sequence is of length 363. The backbone hyperedges are excluded.

window size	$E_P \cap E_T$	$E_P \cap E_T^c$	$E_P^c \cap E_T$
7	69	0	0
8	14	0	0

Tab. 2 shows the result of conformation of protein 4ALD obtained by applying a (1,2)-conformation rule learned from the other 14 TIM-barrel proteins. In the stage of window size 7, 23 (= 6+17) hyperedge are added, 6 hyperedges of which are similar or exactly identical to hyperedges in the target H_T. However, the remaining 17 hyperedges are wrong, that is, there are no similar hyperedges to them in H_T. An interesting observation is that correct hyperedge addition often occurs in a neighborhood, which would imply that the conformation rule causing correct hyperedge addition captures some regional property common to several proteins. In the stage of window size 8, no hyperedge is added. This is because, once a wrong hyperedge is added, the wrong hyperedge makes it difficult to add correct hyperedges in the following stages with larger window sizes. To settle this problem is also a future work.

Table 2. Result of conformation of protein 4ALD applied a (1,2)-conformation rule learned from the other 14 TIM-barrel proteins.

window size	$Sim_{2,0.8}(E_P, E_T)$	$Sim_{2,0.8}(E_T, E_P)$	$E_P \cap E_T^c$	$E_P^c \cap E_T$
7	6	9	17	63
8	0	0	0	14

8 Concluding Remarks

In this paper, we formulated the protein conformation problem as the PAC-learning problem of hypergraph rewriting rules from hypergraphs. Since, in terms of the protein conformation problem, our graph-theoretic approach is very unique, this learning problem should be extensively studied with adding appropriate modification to the framework we proposed this time, although the current results of our preliminary computational experiments are far from satisfaction.

Acknowledgments

This work was in part supported by Grant-in-Aid for Encouragement of Young Scientists and Grant-in-Aid for Scientific Research on Priority Areas (C) "Genome Information Science" from MEXT of Japan, and the Research for the Future Program of the Japan Society for the Promotion of Science.

References

1. Church, B.W. and Shalloway, D., Top-down free-energy minimization on protein potential energy landscapes, *Proc. Natl. Acad. Sci. U.S.A.* **98**, 6098–103, 2001.
2. Dill, K.A., Fiebig, K.M. and Chan, H.S., Cooperatively protein-folding kinetics, *Proc. Natl. Acad. Sci. U.S.A.* **90**, 1942–1946, 1993.
3. Hart, W.E. and Istrail, S.C., Robust proofs of NP-hardness for protein folding: general lattices and energy potentials, *J. Comput. Biol.* **4**, 1–22, 1997.
4. Konig, R. and Dandekar, T., Improving genetic algorithms for protein folding simulations by systematic crossover, *Biosystems* **50**, 17–25, 1999.
5. Matsumoto, S. and Shinohara A., Refutably probably approximately correct learning, *Proc. 5th International Workshop on Algorithmic Learning Theory*, LNAI **872**, 469–483, 1994.
6. Matsumoto, S., *Studies on the learnability of pattern languages.* PhD thesis, Kyushu University, 1998.
7. Natarajan, B.K., Probably approximate learning of sets and functions, *SIAM J. Comput.* **20**, 328–351, 1991.
8. Natarajan, B.K., *Machine Learning: A Theoretical Approach*, Morgan Kaufmann, 1991.
9. Natarajan, B.K. and Tadepalli, P., Two new frameworks for learning, *Proc. Fifth International Symposium on Machine Learning*, 402–415, 1988.
10. Shimozono, S., Shinohara, A, Shinohara, T., Miyano, S., Kuhara, S. and Arikawa, S., Knowledge acquisition from amino acid sequences by machine learning system BONSAI, *Trans. Information Processing Society of Japan* **35**, 2009–2018, 1994.
11. Smith, R.F. and Smith, T.F., Automatic generation of primary sequence patterns from sets of related protein sequences, *Proc. Natl. Acad. Sci. U.S.A.* **87**, 118–122, 1990.
12. Wierenga, R.K., The TIM-barrel fold: a versatile framework for efficient enzymes, *FEBS Letters* **492**, 193–198, 2001.
13. http://www.python.org/
14. http://www.rcsb.org/pdb/

Knowledge Navigation on Visualizing Complementary Documents

Naohiro Matsumura[1,3], Yukio Ohsawa[2,3], and Mitsuru Ishizuka[1]

[1] Graduate School of Engineering, University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan
{matumura, ishizuka}@miv.t.u-tokyo.ac.jp
[2] Graduate School of Systems Management, University of Tsukuba,
3-29-1 Otsuka, Bunkyo-ku, Tokyo, 112-0012 Japan
osawa@gssm.otsuka.tsukuba.ac.jp
[3] Japanese Science and Technology Corporation,
2-2-11 Tsutsujigaoka, Miyagino-ku, Sendai, Miyagi, 983-0852 Japan

Abstract. It is an up-to-date challenge to get answers for novel questions which nobody has ever considered. Such a question is too rare to be satisfied with a past single document. In this paper, we propose a new framework of knowledge navigation by graphically providing with multiple documents relevant to a user's question. Our implemented system named MACLOD generates several navigational plans, each forming a complementary document-set, not a single document, for navigating a user to understanding a novel question. The obtained plans are mapped into a 2-dimensional interface where documents in each obtained document-set are connected with links in order to support user selecting a plan smoothly. In experiments, the method obtained satisfactory answers to user's unique questions.

1 Introduction

It is an up-to-date challenge to answer a user's novel question nobody has ever asked. However, such a question is too new to be satisfied with a past single document, and the required knowledge for understanding the documents relevant to a user's question depends on his background[4]. In our previous work[3], we proposed a novel information retrieval method named *combination retrieval* for creating novel knowledge by combining complementary documents. Here, a complementary set of documents is composed of documents, and the combination of which supplies a satisfactory information. This idea is based on the principle that combining ideas can trigger the creation of new ideas[1,2]. Throughout the discussions of the work, we verified the fact that reading multiple complementary documents generates the synergy effects which help us acquire novel knowledge.

In this paper, we propose a new framework of knowledge navigation, i.e., supply a user with new knowledge, for satisfying the information request of a user by visualizing complementary documents. Our implemented system named

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 258–270, 2001.

MACLOD(Map of Complementary Links of Documents) generates several navigational plans, each formed by a document-set for navigating a user to understand a novel question, by making use of the combination retrieval[3]. The obtained plans are mapped into a 2-dimensional interface where documents in each document-set are connected with links in order to support user selecting complementary documents smoothly.

The remainder of this paper goes as follows: In Section 2, the meaning of our approach is shown by comparison with previous knowledge navigation methods. The mechanism of combination retrieval is described in Section 3, and the mechanism of MACLOD implemented here is described in Section 4. We show the experiments and the results in Section 5, showing the performance of MACLOD for medical counseling question-answer documents.

2 Previous Methods for Knowledge Navigation

The vision of knowledge navigation was shown by John Sculley(Then the president of Apple Computer Inc.) where electronic secretary in a computer named Knowledge Navigator managed various tasks on behalf of users, e.g., manage schedules. The concept inspired us. However, it is still difficult to realize the Knowledge Navigator because of the complexity of real secretary's tasks.

A knowledge navigation system is a piece of software which answers a user's question. The question maybe entered as a word-set query $\{alcohol, liver, cancer\}$ or a sentence query "Does alcohol cause a liver cancer ?" An intelligent answer to this question may be "No, alcohol does not cause liver cancer directly. You may be confused of liver cancer and other liver damages from alcohol. Alcohol causes cancer in other tissues." For giving such an answer, the system should have medical knowledge relevant to user's query, and infer on the knowledge for answering the question. However, it is not realistic to implement such knowledge wide enough to be applied to unique user interests.

Another approach for navigating knowledge is to retrieve ready-made documents relevant to the current query, from a prepared document collection. In this way, we can skip the process of knowledge acquisition and implementation, because man-made documents represent the complex human knowledge directly. A search engines for a word-set query entered by the user may be the simplest realization of this approach. However, we already know that existing information retrieval methods trying to answer a query by ONE of the output documents could not satisfy novel interests in Section 1.

3 The Process of Combination Retrieval

Combination retrieval[3] is a method for selecting meaningful documents which, as a set, serve a good (readable and satisfactory) answer to the user. In this section, we review the algorithm of the combination retrieval.

3.1 The Outline of the Process

The process of combination retrieval is as follows:

The Process of Combination Retrieval
Step 1) Accept user's query Q_g.
Step 2) Obtain G, a word-set representing the goal user wants to understand, from Q_g ($G = Q_g$ if Q_g is given simply as a word-set).
Step 3) Make knowledge-base Σ for the abduction of Step 4). For each document D_x in the document-collection C_{doc}, a Horn clause is made as to describe the condition (words needed to be understood for reading D_x) and the effect (words to be subsequently understood by reading D_x).
Step 4) Obtain h, the optimal hypothesis-set which derives G if combined with Σ, by cost-based abduction (detailed later). h obtained here represents the union of following information, of the least size of K.
S**:** The document-set the user should read.
K**:** The keyword-set the user should understand for reading the documents in S.
Step 5) Show the documents in S to the user.

The intuitive meaning of employing the abductive inference is to obtain the conditions for understanding user's goal G. Here, conditions include the documents to read (S) for understanding G, and necessary knowledge (K) for reading those documents. That is, S means the combination of documents to be presented to the user.

3.2 The Details of Combination Retrieval's Process

In preparation, collection C_{doc} of existing human-made documents is stored. Key, the set of keyword-candidates in the documents in C_{doc}, i.e. word-set which is the union of extracted keywords from all the documents in C_{doc}, is obtained and fixed. Here, words are stemmed as in [5] and stop words ("does", "is", "a"...) are deleted, and then a constant number of words of the highest TFIDF values [6] (using C_{doc} as the corpus for computing document frequencies of words) are extracted as keywords from each document in C_{doc}. Next, let us go into the details of each step in 3.1.
Step 1) to 2) Make goal G from user's query Q_g: Goal G is defined as the set of words in $Q_g \cap Key$, i.e., keywords in the user's query. For example, "does alcohol make me warm?" and query $\{alcohol,\ warm\}$ are both put into the same goal $\{alcohol,\ warm\}$, if C_{doc} is a set of past question-answer pairs of a medical counselor which do not have "does", "make", "me", "warm", "in", "a", or "day" in Key (some are deleted as stop words).
Step 3) Make Horn clauses from documents: For the abductive inference in Step 4) of Subsection 3.1, knowledge-base Σ is formed of *Horn clauses*. A Horn clause is a clause as in Eq.(1), which means that y becomes true under the condition that all $x_1, x_2, \cdots x_n$ are true, where variables $x_1, x_2, \cdots x_n$ and y

are atoms each of which corresponds to an event occurrence. A Horn clause can describe causes $(x_1, x_2, \cdots, x_n)$ and their effect (y) simply.

$$y :- x_1, x_2, \cdots, x_n. \tag{1}$$

In combination retrieval, the Horn clause for document D_x describes the cause (reading D_x with enough vocabulary knowledge) and the effect (acquiring new knowledge from D_x) of reading D_x, as:

$$\alpha :- \beta_1, \beta_2, \cdots, \beta_{mx}, D_x. \tag{2}$$

Here, α is the *effect term* of D_x, which is a term (a word or a phrase) one can understand by reading document D_x. $\beta_1,\ \beta_2 \cdots \beta_{mx}$ are the *conditional terms* of D_x, which should be understood for reading and understanding D_x. That is, one who knows words $\beta_1,\ \beta_2 \cdots \beta_{mx}$ and reads D_x on this knowledge is supposed to acquire knowledge about α.

The method for taking the effect and the conditional terms from D_x is straight-forward. First, the effect terms $\alpha, \alpha_2, \cdots$ are obtained as terms in $G \cap (\textit{the keywords of } D_x)$. This means that the effect of D_x is expected on the user's interest G, rather than by the intension of the author of D_x. For example, a document about cancer symptoms may work as a description of the demerit of smoking, if the reader is a heavy smoker. Focusing the consideration onto user's goal in this way also speeds up the response of combination retrieval as in Subsection 5.1.

Then, the keywords of D_x other than the effect terms above form the conditional terms $\beta_1,\ \beta_2,\ \cdots \beta_{mx}$. As a result, Horn clauses are obtained as

$$\begin{aligned} \alpha_1 &:- \beta_1, \beta_2, \cdots \beta_{mx}, D_x, \\ \alpha_2 &:- \beta_1, \beta_2, \cdots \beta_{mx}, D_x, \\ &\vdots \end{aligned} \tag{3}$$

meaning that one knowing $\beta_1,\ \beta_2, \cdots \beta_{mx}$ can read D_x and understand all the effect terms $\alpha_1,\ \alpha_2, \cdots$ by reading D_x.

Step 4) Cost based abduction for obtaining the documents to read: We employ the *cost based abduction* (CBA, hereafter)[7], an inference framework for obtaining solution h of the least $|K|$ in Subsection 3.1. In CBA, the causes of a given effect G is explained. Formally, CBA is described as extracting a minimal hypothesis-set h from a given set H of candidate hypotheses, so that h derives G using knowledge Σ. That is, h satisfies Eq.(4) under Eq.(5) and Eq.(6). We deal with Σ composed of causal rules, expressed in Horn clauses mentioned above.

$$Minimize\ cost(h),\ under\ that: \tag{4}$$

$$h \subset H, \tag{5}$$

$$h \cup \Sigma \vdash G, \tag{6}$$

Eq.(4) represents the selection of h to be minimal, i.e., of the lowest-cost hypothesis-set $h(\subset H)$, where cost denoted by $cost(h)$ is the sum of the *weights* of hypotheses in h. The weights of hypotheses in H, the candidates of elements of solution h, are initially given. Generally speaking, the weight-values of hypotheses are closely related to the semantics in the problem to which CBA is applied, as exemplified in [8]. In combination retrieval, weights are given differently to the two types of hypotheses in H:

Type 1: Hypothesis that user reads a document in C_{doc}
Type 2: Hypothesis that user knows (have learned) a conditional term in Key

In giving weights to hypotheses, we considered that user should be able to understand the output documents in S, with learning only a small set K of keywords from external knowledge other than C_{doc}. This is reflected to minimizing $|K|$, the size of K. That is, the weights of hypotheses of Type 2 are fixed to 1 and ones of Type 1 are fixed to 0, and the content of h is $S \cup K$. It might be good to give values between 0 and 1 to hypotheses of Type 2, each value representing the difficulty of learning each term. However, we do not know how each word is easy to learn for the user from outside of C_{doc}. Further, it might seem to be necessary to give positive weights to hypotheses of Type 1, each value representing the cost of reading each document. However, this necessity can be discounted because we gave mx in Eq. 3 to be proportional to the length of D_x. That is, the user's cost (effort) for reading a document is implied by the number of meaningful keywords s/he should read in the document. If we sum the heterogeneous difficulties, i.e., of reading documents and of learning words, the meaning of the solution cost would become rather confusing.

3.3 An Example of Combination Retrieval's Execution

For example, the combination retrieval runs as follows.

Step 1) Q_g = "Does alcohol cause a liver cancer ?"
Step 2) G is obtained from Q_g as $\{alcohol,\ liver,\ cancer\}$.
Step 3) From C_{doc}, documents D_1,D_2, and D_3 are taken, each including terms in G, and put into Horn clauses as:

$$alcohol :- cirrhosis, cell, disease, D_1.$$
$$liver :- cirrhosis, cell, disease, D_1.$$
$$alcohol :- marijuana, drug, health, D_2.$$
$$liver :- marijuana, drug, health, D_2.$$
$$alcohol :- cell, disease, organ, D_3.$$
$$cancer :- cell, disease, organ, D_3.$$

Hypothesis-set H is formed of the conditional parts of D_1, D_2 and D_3 of Type 1 each weighted 0, and "cirrhosis," "cell," "disease," "marijuana," "drug," "health,"and "organ" of Type 2 each weighted 1.

Step 4) h is obtained as $S \cup K$, where

$$S = \{\ D_1,\ D_3\}\ and$$
$$K = \{cirrhosis,\ cell,\ disease,\ organ\},$$

meaning that user should understand "cirrhosis", "cell", "disease" and "organ" for reading D_1 and D_3, served as the answer to Q_g. This solution is selected because $cost(h)$ (i.e. $|K|$) takes the values of 4, less than 6 of the only alternative feasible solution, i.e. $\{marijuana,\ drug,\ health,\ cell,\ disease,\ organ\}$ plus $\{D_2,\ D_3\}$.

Step 5) The user now reads the two documents presented as:

D_1 (including *alcohol* and *liver*) stating that alcohol alters the liver function by changing liver cells into cirrhosis.

D_3 (including *alcohol* and *cancer*) showing the causes of cancer in various organs, including a lot of alcohol. This document recommends drinkers to limit to one ounce of pure alcohol per day.

As a result, the subject learns that s/he should limit drinking alcohol to keep liver healthy and avoid cancer, and also came to know that other tissues than liver get cancer from alcohol.

Thus, user can understand the answer by learning a small number of words from outside of C_{doc}, as we aimed in employing CBA. More importantly than this major effect of combination retrieval, a by-product is that the common hypotheses between D_1 and D_3, i.e., $\{cell,\ disease\}$ of Type 2 are discovered as the context of user's interest underlying the entered query. This effect is due to CBA which obtains the smallest number of involved contexts, for explaining the goal (i.e. answering the query), as solution hypotheses. Presenting such a novel and meaningful context to the user induces the user to creating new knowledge [9], to satisfy his/her novel interest.

4 MACLOD: Map of Complementary Links of Documents

In the combination retrieval, a user was imposed on two types of tasks that reading a obtained document-set and understanding the conditional terms of the document-set. However, this tasks are not always easy for a user since the background knowledge of a user is different from individuals. For taking such already existing knowledge of a user into consideration when generating the document-set for reading, we propose a new framework to navigate a user by graphically providing with multiple documents of some document-sets each giving an answer to his/her interest. The concept of knowledge navigation in Section 2 can be realized in the framework.

The implemented system named *MACLOD* (MAp of Complementary Links Of Documents) visualizes these document-sets(each forming a complementary document-set) to navigate a user to understanding his/her novel question. The process of MACLOD is as follows:

The Process of MACLOD

Phase1. Obtain a plan for knowledge navigation: Obtain a plan (document-set S) for user's query Q_g along the procedure of the combination retrieval in Section 3. That is, the process is summarized as follows:

Step 1) Accept user's query Q_g.
Step 2) Obtain G, the goal user wants to understand.
Step 3) Make knowledge-base Σ for the abduction of Step 4).
Step 4) Obtain h, the optimal hypothesis-set which derives G if combined with Σ, by cost-based abduction.
Step 5) Show the obtained documents in Step 4) to the user.

Phase2. Iterate Phase1 to add plans: Iterate Phase1 to obtain N sets of plans where inconsistency conditions are added to the knowledge-base Σ in Subsection 3.2 to avoid already obtained plans. The inconsistency condition to be considered in each cycle of Phase1 is described as

$$inc :- D_{x1}, D_{x2}, \cdots, D_{xn}, \tag{7}$$

where $D_{x1}, D_{x2}, \cdots, D_{xn}$ are the documents obtained in the previous cycle of Phase1. Here, the document included in S more than three times are forced not to be included in the next plan. This inconsistency condition, also added into knowledge-base Σ, is described as

$$inc :- D_{x1}. \tag{8}$$

Where D_{x1} is a document included in S more than three times. The cycles of **Phase1** continues until the number of iterations reaches N. Here, we empirically set N as 10.

Phase3. Visualize the plans: MACLOD outputs a 2-dimensional interface in which obtained plans during above iterations are mapped. In the interface, documents in a plan obtained by one cycle at Phase1 are connected with links each other in order to support user selects appropriate documents.

Phase4. Knowledge Navigation: The user goes on reading documents along the links in the 2-dimensional interface until s/he understands or gives up understanding Q_g.

5 Experimental Evaluations of MACLOD

5.1 The Experimental Conditions

MACLOD is implemented in a Celeron 500MHz machine with 320MB memory. Although CBA is time-consuming because of its NP-completeness, most answers in the experiments were returned within ten seconds from the entry of query by high-speed abduction as in [12]. Queries from users included 4 or less terms in Key, due to which the response time was below 10 sec. This quick response comes also from the goal-oriented construction of Horn clauses shown in Subsection 3.2. The document-collection C_{doc} of MACLOD is

1808 question-answer pairs of *Alice*, a health care question answering service on WWW (`http://www.alice.columbia.edu`). The small number as 1808 documents is a suitable condition for evaluating MACLOD for a sparse document-collection which is insufficient for answering novel queries.

5.2 An Example of MACLOD's Execution

When a user entered a query in a word-set or a sentence, MACLOD obtained ten plans(document-sets) in Table 1 and showed a 2-dimensional output in Figure 5.2. In this case, input $\{alcohol,\ fat,\ calorie\}$ was entered as query Q_g for knowing if the calorie of alcohol changes into fat.

Table 1. The top 10 plans for the input query {alcohol, fat, calorie}.

Ranking	*Plan(document-set)*	*Cost*
1	$d1459$, $d0181$	25
2	$d1459$, $d0611$	26
3	$d1459$, $d0426$	27
4	$d1802$, $d0181$	27
5	$d0576$, $d0181$	27
6	$d1802$, $d0882$, $d0611$	39
7	$d1802$, $d1100$, $d0611$	39
8	$d0746$, $d0576$, $d1466$	39
9	$d1730$, $d0576$, $d1466$	39
10	$d0746$, $d0331$, $d1466$	41

The process of understanding the user's interest(shown as Q_g) begins by reading a document-set $d1459$ and $d0181$ (double-circle nodes in Figure 5.2), a top ranked plan of MACLOD. The summaries of them are as follows:

$d1459$ (including fat and $calorie$) stating that if the calorie comes short, the protein is burned into energy. The lack of protein delays the recovery of distress, or weakens the resistance to disease.

$d0181$ (including $alcohol$) stating that drinking too much alcohol damages various tissues, especially the liver or the heart.

After reading these two documents, the user was not satisfy fully his/her interest since the documents do not mention the causality between the calorie of alcohol and fat directly. If this does not satisfy one's interest, then the user begins to select and read another documents linked from already read documents for getting new information about Q_g. MACLOD helps this complementary reading process with a 2-dimensional interface where a user can piece out the whole relations among documents of obtained plans. That is, user can pick other document, that complements already-read documents, for reaching the satisfaction of her/himself.

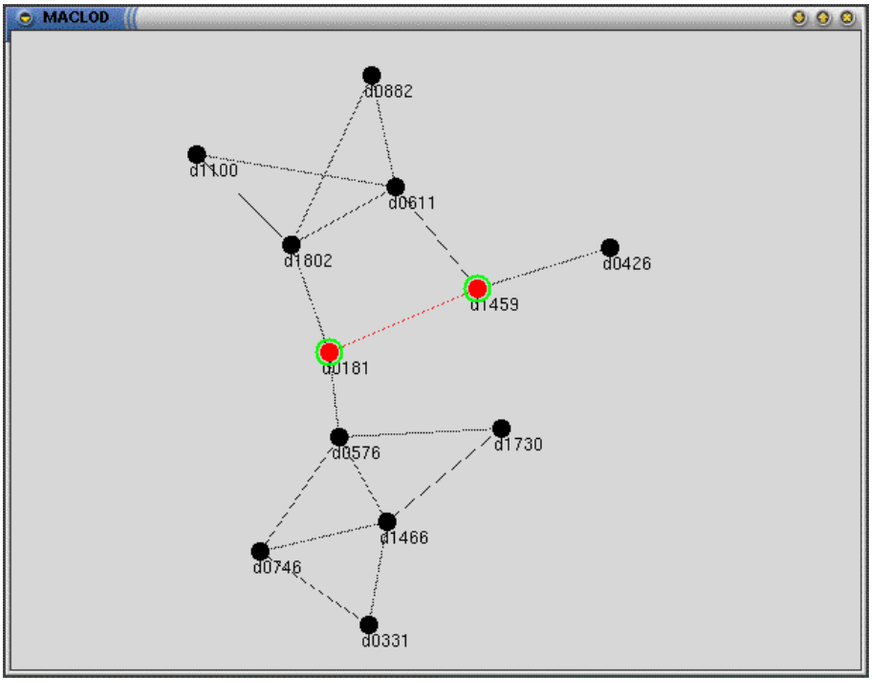


Fig. 1. A 2-dimensional interface of MACLOD. Documents are shown as nodes, and complementary documents are connected with links.

The following steps, for example, are as follows. In Figure 5.2, $d0611$ and $d0426$ are linked from $d1459$, and $d1802$ and $d0576$ are linked from $d0181$. Here, because the user wanted to know the limit amount of alcohol to drink, the user was satisfied by reading $d0611$ that states the adequate quantity of alcohol per day. Also, $d0576$ stating the ideal quantity of calorie per day satisfied the user further because his potential interest was in diet. Thus, MACLOD can supply complementary documents step by step according to the user's interests until the user gets satisfied.

5.3 The Answering System Compared with MACLOD

We compared the performance of MACLOD with the following typical search engine for question answering. We call this search engine here a Vector-based FAQ-finder (*VFAQ* in short hereafter).

The Procedure of VFAQ
Step1') Prepare keyword-vector v_x for each question Q_x in C_{doc}.
Step2') Obtain keyword-vector v_Q for the current query Q_g.

Step3') Find the top N' keyword-vectors prepared in 1'), in the decreasing order of product value $v_x \cdot v_Q$, and return their corresponding answers.

Here, a keyword-vector for a query Q is formed as follows: Each vector has $|Key|$ attributes (Key was introduced in 3.2 as the candidate of keywords in C_{doc}), each taking the value of TFIDF[6] in Q, of the corresponding keyword. Each vector v is normalized to make $|v| = 1$. For example, for query Q_g $\{alcohol,\ warm\}$ (or a question which is put into G: $\{alcohol,\ warm\}$), the vector comes to be $(0, 0.99, 0, \cdots, 0, 0.14, 0, 0, \cdots)$ where 0.99 and 0.14 are the normalized TFIDF values of "alcohol" and "warm" in Q_g. Elements of value 0 here correspond to terms which are in Key but not included in Q_g. Supplying N' documents in Step 3') is for setting the condition similar to MACLOD so that a fair comparison becomes possible.

5.4 Result Statistics

The experiment was executed for 5 subjects from 21 to 30 years old. This means that the subjects were of the near age to the past question askers of *Alice*.

A popular method for evaluating the performance of a search engine is to see *recall* (the number of relevant documents retrieved, divided by the number of relevant documents to user's query in C_{doc}) and *precision* (the number of relevant documents retrieved, divided by the number of retrieved documents). However, this traditional manner of evaluation is not appropriate for evaluating MACLOD, because it does not output a sheer list of most relevant documents to the query. In the traditional evaluation, it was regarded as a success if user gets satisfied by reading a few documents which are highly ranked in the output list. On the other hand, MACLOD aims at satisfying a user who reads some documents along the pathways, rather than a few best document. Therefore, this section presents an original way of evaluation for MACLOD.

Here, 42 queries were entered. This seems to be quite a small number for the evaluation data. However, we compromised with this size of data because we aimed at having each subject evaluate the returned answer in a natural manner. That is, in order to have the subject report whether s/he was really satisfied with the output, the subject must enter his/her real r̈areïnterest. Otherwise, the subject has to imagine an unreal person who asks the rare query and imagine what the unreal person feels with the returned answers. Therefore we restricted to a small number of queries entered from real novel interests.

The overall result was shown in Figure 5.4. The horizontal axis means the number of documents read in series and the vertical axis means the number of satisfied queries. According to the subjects, MACLOD did better than VFAQ, especially for novel queries. For $x = 1$, MACLOD and VFAQ equally satisfied 16 queries. On the other hand, for $x = 2$, MACLOD satisfied 12 queries, whereas VFAQ satisfied 4 queries. And for $x = 3$, MACLOD satisfied 6 queries, whereas VFAQ satisfied 3 queries. Finally, fot $x \geq 4$, MACLOD and VFAQ satisfied 3 queries. Thus, the superiority of MACLOD for x greater than 1 came to be

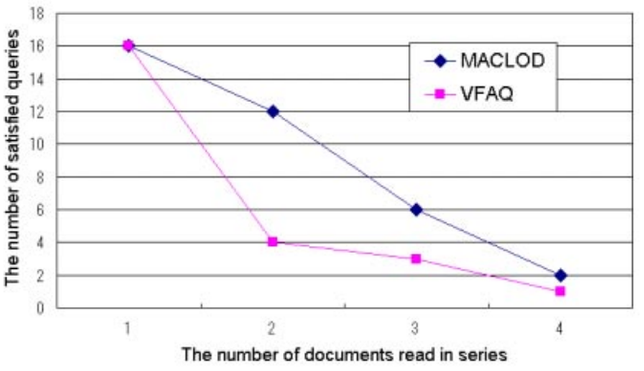


Fig. 2. Statistical results.

apparent. In all cases, VFAQ obtained redundant documents, i.e., document of similar contexts, equally relevant to the query.

These results can be summarized that novel queries for C_{doc} were answered satisfactory by MACLOD. Answers in the form of document-combination visualized by MACLOD came to be easy to read and browse along the links according to the subject, and the presented answers were meaningful for the user.

5.5 Comparison with Other Methods

Among the rare systems which combine documents for answering novel query, Hyper Bridges[10] and *NaviPlan* [11] produce a plan of user's reading of documents. They present a plan made of sorted multiple documents, and a user who reads them in the order as sorted by Hyper Bridges or *NaviPlan* incrementally refines one's own knowledge until one learns the meaning of the entered query. A plan made by these tools is a *serial* set of documents, which guides the user to an understanding of query starting from a beginner's knowledge, in the order presented by the system. As a result, neither *NaviPlan* nor Hyper Bridges they can obtain an appropriate document to be read last, i.e., the document to directly reach the goal (i.e. answer the query), in all the examples above where multiple documents are required to be mixed to answer the query. On the other hand, the combination retrieval and its advanced version MACLOD makes a *complementary* set of documents, supplementing the content of each other for giving a satisfactory answer as a whole. User may read documents in an obtained document-set in any order as s/he likes. Especially, MACLOD gives user more flexible search interface than the original combination retrieval.

Let us here show the merit of MACLOD compared with the previous combination retrieval. In short, the merit is that user can select documents matching with his/her interest, reactively reflecting the context of documents read already.

The fair extension of the combination retrieval to be compared with MACLOD is to have as many document-sets as obtained in MACLOD. In such an output style, it will be difficult to control the context of the documents to read. That is, the order of sets sorted on *cost* does not always correspond to user's interest and often bothers user with hard to read the document-sets in an undesired order. In this example, if the user feels d1459 mismatching to his/her context, s/he will not reach any satisfactory document-set in the list. Neither a MACLOD-like style output as in Figure5.2 makes things better, in this case because d1459 is shared by all the sets. In all trials for obtaining and showing highly ranked document-sets of the combination retrieval, the user was fixed to the context bound by a c̈entrald̈ocument as d1459 whether desiring or not the situation. From this problem with the combination retrieval, we can point the two-fold merit of MACLOD.

1. Due to discarding documents already appeared many times in the output document-sets in the process (see Section 4), MACLOD can include document-sets of various contexts in the output. This enables the user to choose suitable contexts reactively in the search process.
2. The graphical output makes the context-control easier, because the links between nodes (documents) represent the complemantary relations (i.e., as documents to be read together) between contexts. If user feels a document misleading to him, s/he can open a document linked from the current document without feeing a sudden departure from the current context.

6 Conclusions

The combination retrieval, a method to obtain a set of documents for answering a novel query is fully described and its visual interface MACLOD is introduced. Combination retrieval presents user with a set of, not a single, documents for answering a new query unable to be answered by one past answer to a past query. The MACLOD interface supplies a user with a further comfort in acquiring novel knowledge. MACLOD allows user to efficiently alter a part of the reading-plan (i.e. document-set) s/he is currently following, improving his/her satisfaction. This effect works especially if the interest is novel i.e., if the context is too particular to be captured by past Q&A's.

References

1. Hadamard, J: *The Psychology of Invention in the Mathematical Field*. Princeton University Press, 1945.
2. Swanson, D.R., Smalheiser, N.R.: An Interactive System for Complementary Literatures: a Stimulus to Scientific Discovery. *Artificial Intelligence*, Vol. 91, 183–203, 1997.
3. Matsumura, N., and Ohsawa, Y.: Combination Retrieval for Creating Knowledge from Sparse Document Collection, *Proc. of Discovery Science*, 320–324, 2000.

4. Brookes, B. C.: The foundations of information science, *Journal of Information Science*, 2, 125–133, 1980.
5. Porter, M.F.: An Algorithm for Suffix stripping. *Automated Library and Information Systems*, Vol.14, No.3, 130–137, 1980.
6. Salton, G. and Buckey, C.: Term-Weighting Approach in Automatic Text Retrieval, *Reading in Information Retrieval*, 323–328, 1998.
7. E. Charniak and S.E. Shimony: Probabilistic Semantics for Cost Based Abduction. *Proc. of AAAI-90*, 106–111, 1990.
8. Ohsawa, Y. and Yachida, M.: An Index Navigator for Understanding and Expressing User's Coherent Interest, *Proc. of IJCAI-97*, 1: 722–729, 1997.
9. Nonaka,I. and Takeuchi, H.: *The Knowledge Creating Company*, Oxford University Press, 1995.
10. Ohsawa, Y., Matsuda, K. and Yachida, M.: Personal and Temporary Hyper Bridges: 2-D Interface for Undefined Topics, *J. Computer Networks and ISDN Systems*, 30: 669–671, 1998.
11. Yamada, S. and Osawa, Y.: Planning to Guide Concept Understanding in the WWW. *AAAI-98 Workshop on AI and Data Integration*, 121–126, 1998.
12. Ohsawa, Y. and Ishizuka, M.: Networked Bubble Propagation: A Polynomial-time Hypothetical Reasoning Method for Computing Near-optimal Solutions, *Artificial Intelligence*, Vol.91, 131–154, 1997.

KeyWorld: Extracting Keywords from a Document as a Small World

Yutaka Matsuo[1,2], Yukio Ohsawa[2,3], and Mitsuru Ishizuka[1]

[1] University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, JAPAN,
matsuo@miv.t.u-tokyo.ac.jp,
http://www.miv.t.u-tokyo.ac.jp/~matsuo/
[2] Japan Science and Technology Corporation, Tsutsujigaoka 2-2-11, Miyagino-ku, Sendai, Miyagi, 983-0852, JAPAN,
[3] University of Tsukuba, Otsuka 3-29-1, Bunkyo-ku, Tokyo 113-0012, JAPAN,

Abstract. The small world topology is known widespread in biological, social and man-made systems. This paper shows that the small world structure also exists in documents, such as papers. A document is represented by a network; the nodes represent terms, and the edges represent the co-occurrence of terms. This network is shown to have the characteristics of being a small world, i.e., nodes are highly clustered yet the path length between them is small. Based on the topology, we develop an indexing system called *KeyWorld*, which extracts important terms by measuring their contribution to the graph being small world.

1 Introduction

Graphs that occur in many biological, social and man-made systems are often neither completely regular nor completely random, but have instead a "small world" topology in which nodes are highly clustered yet the path length between them is small [11][10]. For instance, if you are introduced to someone at a party in a small world, you can usually find a short chain of mutual acquaintances that connects you together. In the 1960s, Stanley Milgram's pioneering work on the small world problem showed that any two randomly chosen individuals in the United States are linked by a chain of six or fewer first-name acquaintances, known as "six degrees of separation" [5]. Watts and Strogatz have shown that a social graph (the collaboration graph of actors in feature films), a biological graph (the neural network of the nematode worm *C. elegans*), and a man-made graph (the electrical power grid of the western United States) all have a small world topology [11][10]. The World Wide Web also forms a small world network [1].

In the context of document indexing, an innovative algorithm called *KeyGraph* [6] is developed, which utilizes the structure of the document. A document is represented as a graph, each node corresponds to a term,[1] and each edge corresponds to the co-occurrence of terms. Based on the segmentation of this graph

[1] A term is a word or a word sequence.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 271–281, 2001.

into clusters, *KeyGraph* finds keywords by selecting the term which co-occurs in multiple clusters. Recently, *KeyGraph* has been applied to several domains, from earthquake sequences [7] to register transaction data of retail stores, and showed remarkable versatility.

In this paper, inspired by both small world phenomenon and *KeyGraph*, we develop a new algorithm, called *KeyWorld*, to find important terms. We show first that the graph derived from a document has the small world characteristics. To extract important terms, we find those terms which contribute to the world being small. The contribution is quantitatively measured by the difference of "small-worldliness" with and without the term.

The rest of the paper is organized as follows. In the following section, we first detail the small world topology, and show that some documents actually have small world characteristics. Then we explain how to extract the important terms in Section 3. We evaluate *KeyWorld* and suggest further improvements in Section 4. Finally, we discuss future works and conclude this paper.

2 Term Co-occurrence Graph and Small World

2.1 Small-Worldliness

We treat an *undirected*, *unweighted*, *simple*, *sparse* and *connected* graph. (We expand to an *unconnected* graph in Section 3.) To formalize the notion of a small world, Watts and Strogatz define the clustering coefficient and the characteristic path length [11][10]:

- The *characteristic path length*, L, is the path length averaged over all pairs of nodes. The path length $d(i, j)$ is the number of edges in the shortest path between nodes i and j.
- The *clustering coefficient* is a measure of the cliqueness of the local neighborhoods. For a node with k neighbors, then at most ${}_kC_2 = k(k-1)/2$ edges can exist between them. The clustering of a node is the fraction of these allowable edges that occur. The clustering coefficient, C is the average clustering over all the nodes in the graph.

Watts and Strogatz define a small world graph as one in which $L \geq L_{rand}$ (or $L \approx L_{rand}$) and $C \gg C_{rand}$ where L_{rand} and C_{rand} are the characteristic path length and clustering coefficient of a random graph with the same number of nodes and edges. They propose several models of graphs, one of which is called β-Graphs. Starting from a regular graph, they introduce disorder into the graph by randomly rewiring each edge with probability p as shown in Fig.1. If $p = 0$ then the graph is completely regular and ordered. If $p = 1$ then the graph is completely random and disordered. Intermediate values of p give graphs that are neither completely regular nor completely disordered. They are small worlds.

Walsh defines the proximity ratio

$$\mu = (C/L) \,/\, (C_{rand}/L_{rand}) \tag{1}$$

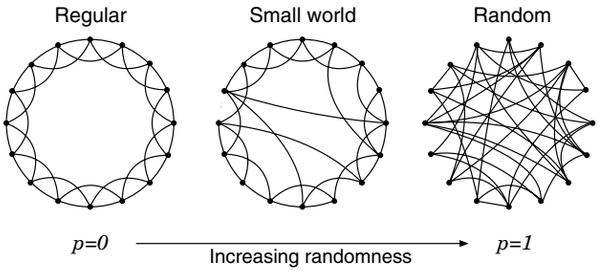


Fig. 1. Random rewiring of a regular ring lattice.

Table 1. Characteristic path lengths L, clustering coefficients C and proximity ratios μ for graphs with a small world topology [9] (studied in [11])).

	L	L_{rand}	C	C_{rand}	μ
Film actor	3.65	2.99	0.79	0.00027	2396
Power grid	18.7	12.4	0.080	0.005	10.61
C. elegans	2.65	2.55	0.28	0.05	4.755

The graphs are defined as follows. For the film actors, two actors are joined by an edge if they have acted in a film together. For the power grid, nodes represent generators, transformers and substations, and edges represent high-voltage transmission lines between them. For *C. elegans*, an edge joins two neurons if they are connected by either a synapse or a gap junction. Because the number of nodes and edges for each graph is different, the magnitude of L, C and μ differs.

as the small-worldliness of the graph [9]. As p increases from 0, L drops sharply since a few long-range edges introduce short cuts into the graph. These short cuts have little effect on C. As a consequence the proximity ratio μ rises rapidly and the graph develops a small world topology. As p approaches 1, the neighborhood clustering starts to break down, and the short cuts no longer have a dramatic effect at linking up nodes. C and μ therefore drop, and the graph loses its small world topology. In Table 1, we can see μ is large in the graphs with a small world topology.

In short, small world networks are characterized by the distinctive combination of high clustering with short characteristic path length.

2.2 Term Co-occurrence Graph

A graph is constructed from a document as follows. We first preprocess the document by stemming and removing *stop words*, as in [8]. We apply an n-gram to count phrase frequency. Then we regard the title of the document, each section title and each caption of figures and tables as a sentence, and exclude all the figures, tables, and references. We get a list of sentences, each of which consists of words (or phrases). In other words, we get basket data where each item is a term, discarding the information of term orderings and document structures.

Table 2. Statistical data on proximity ratios μ for 57 graphs of papers in WWW9.

	L	L_{rand}	C	C_{rand}	μ
Max.	4.99	3.58	0.38	0.012	22.81
Ave.	5.36	—	0.33	—	15.31
Min.	8.13	2.94	0.31	0.027	4.20

We set $f_{thre} = 3$. We restrict attention to the giant connected component of the graph, which include 89% of the nodes on average. We exclude three papers, where the giant connected component covers less than 50% of the nodes. We don't show the L_{rand} and C_{rand} for the average case, because n and k differs dependent on the target paper. On average, $n = 275$ and $k = 5.04$.

Then we pick up *frequent terms* which appear over a user-given threshold, f_{thre} times, and fix them as nodes. For every pair of terms, we count the *co-occurrence* for every sentence, and add an edge if the Jaccard coefficient exceeds a threshold, J_{thre}.[2] The Jaccard coefficient is simply the number of sentences that contain both terms divided by the number of sentences that contain either term. This idea is also used in constructing a referral network from WWW pages [4]. We assume the length of each edge is 1.

Table 2 shows statistics of the small-worldliness of 57 graphs, each constructed from a technical paper that appeared at the 9th international World Wide Web conference (WWW9) [12]. From this result, we can conjecture these papers certainly have small world structures. However, depending on the paper, the small-worldliness varies.

One reason why the paper has a small world structure can be considered that the author may mention some concepts step by step (making the clustering of related terms), and then try to merge the concepts and build up new ideas (making a 'shortcut' of clusters). The author will keep in mind that the new idea is steadily connected to the fundamental concepts, but not redundantly. However, as we have seen, the small-worldliness varies from paper to paper. Certainly it depends on the subject, the aim, and the author's writing style of the paper.

3 Finding Important Terms

3.1 Shortcut and Contractor

Admitting that a document is a small world, how does it benefit us? We try here to estimate the importance of a term, and pick up important terms, though they are rare in the document, based on the small world structure. We consider 'important terms' as the terms which reflect the main topic, the author's idea, and the fundamental concepts of the document.

[2] In this paper, we set J_{thre} so that the number of neighbors, k, is around 4.5 on average.

First we introduce the notion of a *shortcut* and a *contractor*, following the definition in [10].

Definition 1. *The range $R(i,j)$ is the length of the shortest path between i and j in the* absence *of that edge. If $R(i,j) > 2$, then the edge (i,j) is called a* shortcut*.*

Applying the notion of "shortcuts" in terms of nodes, we can get the definition of "contractor."

Definition 2. *If two nodes u and w are both elements of the same neighborhood $\Gamma(v)$, and the shortest path length between them that does not involve any edges adjacent with v is denoted $d_v(u,w) > 2$, then v is said to* contract *u and w, and v is called a* contractor*.*

In our first thought, if $d_v(u,w)$ is large, the corresponding term of contractor v might be interesting, because they bridge the distant notions which rarely appear together. However, such a node sometimes connects the nodes far from the center of the graph, i.e. the main topic of the document. Below we take into account the whole structure of the graph, calculating the contribution of a node to make the world small.

To treat the disconnected graph, we expand the definition of path length (though Watts restricts attention to the giant connected component[3] of the graph).

Definition 3. *An* extended path length *$d'(i,j)$ of node i and j is defined as follows.*

$$d'(i,j) = \begin{cases} d(i,j), & \text{if } (i,j) \text{ are connected,} \\ w_{sum}, & \text{otherwise.} \end{cases} \tag{2}$$

where w_{sum} is a constant, the sum of the widths of all the disconnected subgraphs. $d(i,j)$ is a path length of the shortest path between i and j in a connected graph.

If some edges are added to the graph and some parts of the graph gets connected, $d'(i,j)$ will not increase, unless the length of an edge is negative. Thus $d'(i,j)$ is one of the upper bounds of the path length considering the edges will be added.

Definition 4. *An* extended characteristic path length *L' is an extended path length averaged over all pairs of nodes.*

Definition 5. *L'_v is an extended path length averaged over all pairs of nodes except node v. L'_{G_v} is the extended characteristic path length of the graph without node v.*

[3] A connected component of a graph is a set of nodes such that each node pair has a path. A connected component is called a giant connected component if it contains more than 50% of the nodes in the graph.

Table 3. Frequent terms in this paper.

Term	Frequency
term	39
small	36
world	35
graph	33
small world	27
node	26
document	25
length	20
important	19
paper	18

Table 4. Terms with 10 largest CB_v in this paper.

Term	CB_v	Frequency
small world	4.38	27
contribution	3.11	11
node	2.98	26
list	2.24	8
author	1.36	7
table	1.10	8
important term	0.80	11
show	0.72	6
structure	0.44	7
KeyWorld	0.44	10

In other words, L'_v is the characteristic path length regarding the node v as a corridor (i.e., a set of edges). For example, if v is neighboring u, w, and z, then $(u, w), (u, z), and (w, z)$ are considered to be linked. And L'_{G_v} is the extended characteristic path length assuming the corridor doesn't exist.

Definition 6. *The* contribution*, CB_v, of the node v to make the world small is defined as follows.*

$$CB_v = L'_{G_v} - L'_v \tag{3}$$

We don't pay attention to the clustering coefficient, because adding or eliminating one node affects the clustering coefficient little.

If node v with large CB_v is absent in the graph, the graph gets very large. In the context of documents, the topics are divided. We assume such a term help merge the structure of the document, thus important.

Table 5. Pairs of terms with 10 largest CB_e.

Pair	CB_e
node – contribution	2.97
list – table	1.47
contribution – important term	1.20
table – show	1.10
contribution – structure	0.87
KeyWorld – list	0.87
important term – develop	0.79
network – show	0.72
contribution – make	0.47
author – idea	0.47

3.2 Example

We show the example experimented on this paper, i.e., the one you are reading now.[4] Table 3 shows the frequent terms and Table 4 shows the important terms measured by CB_v. Comparing two tables, the list of important terms includes the author's idea, e.g., "important term" and "KeyGraph," as well as the important basic concept, e.g., "structure," although they are not frequently appeared. However the list of frequent terms simply show the components of the papers, and are not of interest.

We can also measure the contribution of an edge, CB_e, to make the world small, defined similarly as CB_v. However, if we look at the pairs of terms in Table 5, it is hard to understand what they suggest. There are numbers of relations between two terms, so we cannot imagine the relation of the pairs right away.

Lastly, Fig. 2 shows the graphical visualization of the world of this paper. (Only the giant connected component of the graph is shown, though other parts of the graph is also used for calculation.) We can easily point out the terms without which the world will be separated, say "small world" and "comtribution".

4 Evaluation and Improvements

This section describes an evaluation of *KeyWorld* as an indexing system. *KeyWorld* is not merely an indexing system but it provides an understandable graphical representation of the document. However, we restrict attention here to the performance of *KeyWorld* as an indexing tool to compare it with existing indexing techniques such as *tf* and *tfidf*. The *tf* measures simply term frequency. The *tfidf* measure is obtained by using the product of the term frequency and the inverse document frequency[8].[5]

[4] We ignore the effect of *self-reference*; it's sufficiently small.

[5] We use $\log N/n_v$ as *idf*, where N is the number of document collection, and n_v is the number of document which includes term v.

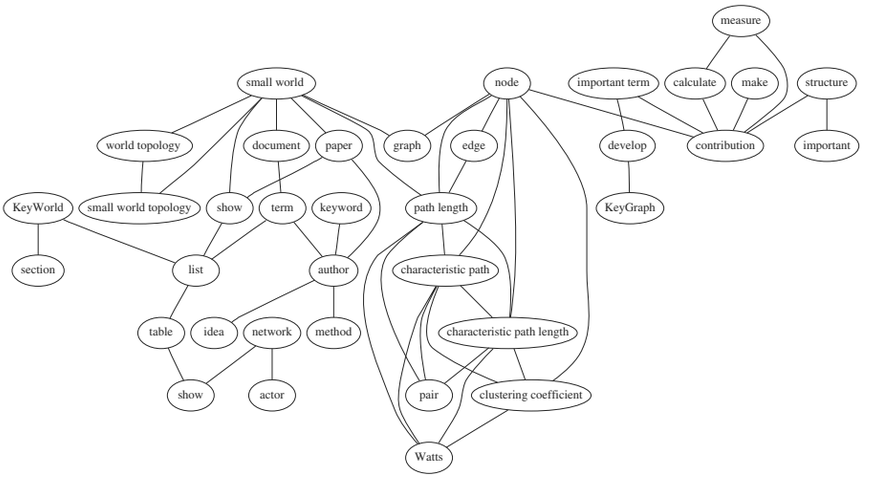


Fig. 2. Small world of this paper.

When an author writes a paper, he/she annotates keywords to his/her paper by selecting the category of the paper (e.g. "text mining"), utilized algorithms (e.g. "small world"), or the proposed method (e.g. "KeyWorld"). The choice depends on the author's criteria. In our definition, a keyword is an important term in the document, which reflects the main topic, the author's idea, and the fundamental concepts of the document. For example, considering this paper, we think "small world," "document," "contribution," "important term," "path length," and "KeyWorld" are keywords, and "node," "make," and "text mining" are not keywords because they are too trivial or too broad, or do not occur in this document.

In the experimentation, we asked the authors of 20 technical papers in the artificial intelligence field to judge whether some terms in their papers are keywords or not by a questionnaire. For each document, we first get top 15 weighted terms by *tf*, *tfidf*,[6] *KeyGraph*, and *KeyWorld*, i.e. the four lists of 15 terms. (We denote the list by method a as $list_a$.) We merge the four lists and shuffle the terms. Then we ask the author whether each term is a keyword or not after explaining the definition of keywords. Counting the number of authorized terms, we can get the precision of method a as follows.

$$precision_a = \frac{\text{Number of authorized terms in } list_a}{\text{Number of terms in } list_a} \tag{4}$$

[6] As a corpus, we used 166 papers in Journal of Artificial Intelligence Research, from Vol.1 in 1993 to Vol.14 in 2001.

Table 6. Precision and coverage

	tf	*KeyWorld*	*tfidf*	*KeyWorld+idf*
precision	0.53	0.49	0.55	0.71
coverage	0.48	0.50	0.62	0.68

Table 7. Terms with 10 largest $CB_v \times idf_v$ in this paper.

Term	$CB_v \times idf_v$	Frequency
small world	4.57	27
important term	3.82	11
co-occurrence	1.89	4
KeyWorld	1.58	10
short cut	1.56	4
actor	0.89	5
shortest path	0.66	4
sentence	0.66	4
document	0.66	23
path length	0.59	17

Next, from the shuffled list of all terms,[7] the authors are told to pick 5 (or more) terms as indispensable terms which they think are essential to the document, and cover the most important concepts of the paper. We calculate the coverage of method a as follows.

$$coverage_a = \frac{\text{Number of indispensable terms in } list_a}{\text{Number of indispensable terms}} \tag{5}$$

The results are shown in Table 6. The performance of *KeyWorld* is not good enough. The precision and coverage are almost equal to *tf*. However, we feel that the term list by *KeyWorld* includes very important terms as well as very dull words, e.g. "show" or "table" in Table 4. To sieve out these dull terms, we develop an improved weighting method, which annotates term v with the weight

$$CB_v \times idf_v, \tag{6}$$

where idf_v is an *idf* measure for term v. The improved results are also shown in Table 6. Both the precision and coverage are now far better than *tfidf*. Table 7 shows the top 10 terms by *KeyWorld* with *idf* factor for this paper.

In summary, *KeyWorld* can often find important terms, however, it also detect less important terms. By incorporating with the *idf* measure, *KeyWorld* can be a very good indexing tool.

[7] If the author remembers the other terms, he/she is permitted to add them to the list.

5 Discussion

The small world phenomenon was inaugurated as an area of experimental study in the social sciences by Stanley Milgram in the 1960's. Since then, numerous studies have been done for network analysis. The importance of weak ties, which is a short cut between clusters of people, was mentioned 30 years ago [3].

The measure of contribution is similar to "*centrality*" in the context of social network study. Centrality can be measured in a number of ways [2]. Considering an actors' social network, the simplest is to count the number of others with whom an actor maintains relations. The actor with the most connections, i.e., the highest *degree*, is most central. Another measure is *closeness*, which calculates the distance from each person in the network to each other person based on the connections among all members of the network. Central actors are closer to all others than are other actors. A third measure is *betweenness* which examines the extent to which an actor is situated between others in the network, i.e., the extent to which information must pass through them to get to others, and thus the extent to which they will be exposed to information circulating in the network. However, our measure of *contribution* has a characteristic in that it calculates the difference of the closeness of all nodes with and without a certain node. It measures a node's contribution to the whole structure by temporarily eliminating the node.

6 Conclusion

Watts mentions in [10] the possible applications of small world research, including "the train of thought followed in a conversation or succession of ideas leading to a scientific breakthrough." In this paper, we have focused on technical papers rather than a conversation or succession of ideas. The future direction of our research is to treat *directed* or *weighted* graphs for finer analyses of documents.

We expect our approach is effective not only to document indexing, but also to other graphical representations. To find out structurally important parts may bring us deeper understandings of the graph, new perspectives, and chances to utilize it. We are interested in a big structural change caused by a small change of the graph. A change, which makes the world very small, may sometimes be very important.

References

1. R. Albert, H. Jeong, and A.-L. Barabasi. The diameter of the World Wide Web. *Nature*, 401, 1999.
2. L. C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1:215–239, 1979.
3. M. Granovetter. Strength of weak ties. *American Journal of Sociology*, 78:1360–1380, 1973.
4. H. Kautz, B. Selman, and M. Shah. The hidden Web. *AI magagine*, 18(2), 1997.

5. S. Milgram. The small-world problem. *Psychology Today*, 2:60–67, 1967.
6. Y. Ohsawa, N. E. Benson, and M. Yachida. KeyGraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In *Proc. Advanced Digital Library Conference (IEEE ADL'98)*, 1998.
7. Y. Ohsawa and M. Yachida. Discover risky active faults by indexing an earthquake sequence. In *Proc. Discovery Science*, pages 208–219, 1999.
8. G. Salton. *Automatic Text Processing*. Addison-Wesley, 1988.
9. T. Walsh. Search in a small world. In *Proc. IJCAI-99*, pages 1172–1177, 1999.
10. D. Watts. *Small worlds: the dynamics of networks between order and randomness*. Princeton, 1999.
11. D. Watts and S. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.
12. 9th International World Wide Web Conference. http://www9.org/.

A Method for Discovering Purified Web Communities

Tsuyoshi Murata

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430 Japan
tmurata@nii.ac.jp

Abstract. Recommendation of representative Web pages of specific topic is important for assisting users' information retrieval from the Web. This paper describes a method for discovering such pages by purifying Web communities using connectivity information of hyperlinks. A complete bipartite of Web graph, which is composed of centers (containing useful information regarding a topic) and fans (containing hyperlinks to centers), can be regarded as a Web community sharing a common interest. The method is based on the assumption that most of the fans contain hyperlinks pointing to representative pages regarding the topic. In the method, both fans and centers are renewed iteratively by the result of the majority vote of the members of previous community. Experimental results show that our method has abilities of finding representative pages for some topics only from a few input URLs.

1 Introduction

The number of Web pages in the world surpasses 2 billion documents as of July 2000. In order to retrieve useful information from such huge Web network, methods for discovering related Web pages are necessary. Although keyword-based search engines are very popular now, they often find difficulty because of the synonymy and the polysemy of natural languages. Several researches of Web mining based on hyperlink information, which is called Web structure mining, are attempted since they have abilities of processing huge amount of Web pages compared with other content-based Web mining approaches.

As Broder pointed out [2], there are the following reasons and goals for the research of Web structure mining:

1. Designing crawl strategies on the Web
2. Understanding of the sociology of content creation on the Web
3. Analyzing the behavior of Web algorithms that make use of link information
4. Predicting the evolution of Web structures such as bipartite cores and webrings, and developing better algorithms for discovering and organizing them
5. Predicting the emergence of important new phenomena in the Web graph

The author has been working on the research of Web structure mining in order to achieve some of the above goals. A Web visualization system [10] shows the relation

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 282-289, 2001.

of input Web pages in the form of graph in which related pages are located close to each other. Another attempt is a Web community discovery system [11] that finds related Web pages based on the assumption that pages composing a complete bipartite graph are regarded as a community sharing a common interest.

This paper focuses on a topic shared by a Web community, and proposes a method for purifying Web communities in order to find representative Web pages regarding the topic of input pages. It often happens that Web surfers who already know some Web pages about specific topic want to find more representative pages about the same topic. Finding representative Web pages is important for assisting users' information retrieval from the Web.

The method proposed in this paper is based on the graph structure of hyperlinks, and it is an extension of Web community discovery method proposed by the author [11]. In this method, the set of Web pages which compose a complete bipartite graph are renewed iteratively by the majority vote of previous members. The procedure is based on the assumption that most of the Web pages that contain hyperlinks pointing to the pages of some specific topic contain hyperlinks to representative pages about the topic. It is expected that such representative pages will be acquired by the iterative majority vote of the members of previous Web communities. The author has developed a system based on this purification method. The system succeeds in finding representative pages for some of the Web communities only from hyperlink information. Sometimes, the system outputs unexpected Web pages that are different from the topic of input Web pages. Such results are shown and analyzed in the section of experimentation.

2 Related Work

As the examples of Web structure mining, which focus on the graph structure of hyperlinks, HITS [7], Web Trawling [9], and PageRank [12] are famous ones. HITS is an algorithm for topic-dependent ranking. In this algorithm, authority and hub are employed as the criteria for evaluating usefulness of each Web page. A hub page on a topic is a page that has hyperlinks to many other pages on that topic, in other words, a page that links to many authorities on the topic. A good authority is a page that is pointed by many good hubs, while a good hub is a page that points to many authorities. For each Web page, authority weights and hub weights are calculated as follows:

1. Sampling step: A focused collection of several thousand Web pages likely to be rich in relevant authorities is generated. First, HITS algorithm constructs a subgraph expected to be rich in relevant authoritative pages, in which it will search for hubs and authorities. To construct this subgraph, the algorithm uses keyword queries to collect a root set of about 200 pages from a traditional index-based search engine. Since many of these pages are presumably relevant to the search topic, some of the pages are expected to contain links to prominent authorities, and others to be linked to by prominent hubs. The root set is therefore expanded into a base set by including all pages that linked to by pages in the root set, and all pages that link

to a page in the root set. Our attention is restricted to this base set for the remainder of this algorithm. This base set typically contains roughly 1000-3000 pages, and a large number of authoritative pages for the search topic are expected to be in this set.

2. Modification step: Hyperlinks between two pages on the same Web site very often serve a purely navigational function, and typically do not represent conferral of authority. All such hyperlinks are deleted from the subgraph induced by the base set, and apply the remainder of the algorithm to this modified subgraph.
3. Weight-propagation step: The algorithm associates a non-negative authority weight x_p and non-negative hub weight y_p with each page p. All x- and y-values are set to a uniform constant initially. (The final results are essentially unaffected by this initialization.) The authority and hub weights are updated as follows: If a page is pointed to by many good hubs, we would like to increase its authority weight. Thus, for a page p, the value of x_p is updated to be the sum of y_p over all pages q that link to p:

$$x_p = \mathrm{Sum}(y_q) \ \{q \text{ such that } q\text{->}p\}$$

where q->p indicates that q links to p. In the same manner, if a page points to many good authorities, its hub weight is increased:

$$y_p = \mathrm{Sum}(x_q) \ \{q \text{ such that } p\text{->}q\}$$

HITS is a simple algorithm based solely on hyperlink information except the acquisition of a root set, and its behavior is analyzed by several researchers. HITS tends to generalize topics that are not sufficiently broad, which is called topic generalization [5]. There are several works for distilling topics of Web communities by using this phenomena [1] [3].

Another point that should be mentioned is that HITS sometimes outputs hubs and authorities which are irrelevant to input topic. When a good hub page of a community contains hyperlinks pointing to pages of several topics, pointed pages irrelevant to input topic may have much authoritative weight and regarded as an authoritative page of the community. Such phenomenon is called topic drift [6]. Another phenomenon is inadvertent topic hijacking [4], when a base set contains a number of Web pages from the same Web site. Since such pages often contain hyperlinks pointing to the same URL (for example, the top page of the site), authority weight of irrelevant pages may be increased.

Several attempts have been made in order to avoid such phenomena, such as using anchor texts and giving weight to hyperlinks[3], and pruning irrelevant pages from base set in advance to the calculation of authority/hub weights[1]. However, it is considered that the fundamental issue of such undesirable behavior of HITS algorithm lies in the generation of base set. In HITS algorithm, base sets are generated by collecting neighboring pages of a root set, which is acquired from the result of keyword-based search engine. The algorithm is based on the assumption that many good authority/hub pages are included in the base set which are generated in the above way. However, this assumption is not always true. Since HITS focuses its attention on the pages of

base set in the process of ranking, its results are heavily dependent on the quality of the base set. On the other hand, Murata's Web community discovery method [11] acquires data in the process of discovery. The goal of the method is to find a complete bipartite graph which is composed of centers (informative pages) and fans (pages containing hyperlinks to centers), and data acquired from a search engine and Web servers are used for renewing the member of centers and fans iteratively. Since the quality of data can be improved by data acquisition in the process of discovery, the method is expected to avoid the above weakness that HITS suffers.

This paper proposes a new method for purification of Web community, which is a modified version of the above method. Members of fans and centers are changed iteratively by a kind of majority vote of each other. In this manner, members of the communities are purified so that representative fans and authorities are acquired.

3 A Method for Purifying Web Communities

A method for discovering Web community [11], which is the base of our new method in this paper, is explained first. The method consists of the following three steps:

1. Search of fans using a search engine
2. Addition of a new URL to centers
3. Repetition of step 1 and step 2

Figure 1 shows the steps of the community discovery method. In the method, some input URLs are accepted as initial centers, and fans which co-refer all of the centers are searched. As shown in the figure, fans are searched from centers by backlink search on a search engine. The next step is to add a new URL to centers based on the hyperlinks included in acquired fans. The fans' HTML files are acquired through the internet, and all the hyperlink contained in the files are extracted. The hyperlinks are sorted in the order of frequency. Since hyperlinks to related Web pages often co-occur, the top-ranking hyperlink is expected to point to a page whose contents are closely related to the centers. Therefore, the URL of the page is added as a new member of centers.

In a method for purifying Web communities, which is newly proposed in this paper, the above steps of renewing fans and centers are modified in the following way:

- If there are few fans which co-refer all the members of centers, one of the members of centers are randomly removed and then corresponding fans are searched by backlink search so that the number of fans will be more than a certain threshold.
- After all the hyperlinks contained in fans' HTML files are extracted, they are sorted in the order of frequency. Then a few URLs of high-ranking hyperlinks are added to the centers and the same numbers of low-ranking URLs that were the members of previous centers are removed from the centers. This means that centers are updated according as the references of corresponding fans. The number of addi-

tion/removal of URLs is limited up to half of the number of centers so that the topic of centers will not change drastically.

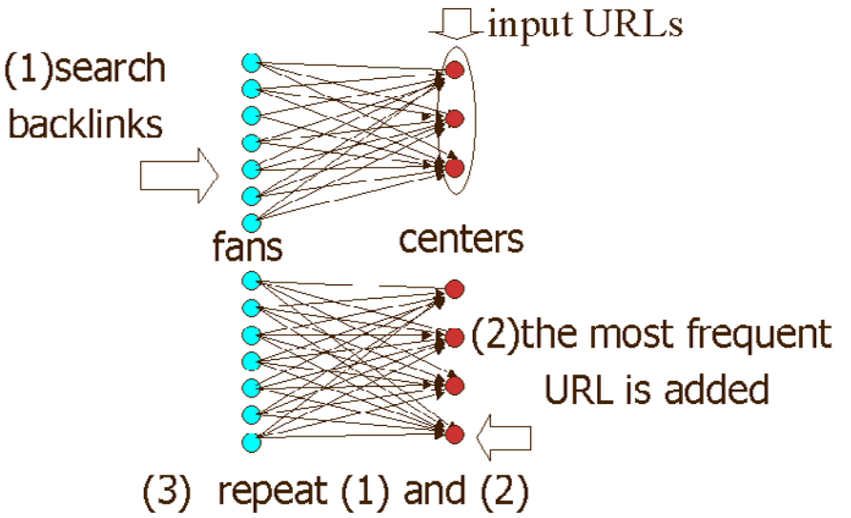


Fig. 1. A method for discovering Web communities

With these modifications, the following effects are expected:

- Even if some irrelevant pages are contained in centers, the quality of fans will not be deteriorated since pages that refer most of the centers (not all of them) are searched and regarded as fans.
- Since the URLs that are linked by many of fans are considered to be representative ones regarding the topic of Web community, replacing the members of centers with high-ranking URLs is expected to improve the quality of centers.

4 Experiments

Based on the above method, the author has built a system for purifying Web communities. As the input to the system, bottom five URLs that are listed in the topics of 100hot.com (http://www.100hot.com/) are given. These URLs are regarded as initial centers of a community, and then it is purified by the system so that higher-ranking URLs will be collected as the members of final centers. Average rankings of centers for each topic before/after purification are shown in Table 1.

This table shows that higher-ranking centers are acquired in some of the topics, such as Macintosh, Election, and Music. The reasons the system performs well for these topics are as follows:

1. Topics of these communities are relatively focused than others. In many cases, there are representative pages that are referred by most of the community members in focused communities.
2. The graphs of these communities are densely connected. This enables the purification only from hyperlink information.

Table 1. Average ranking of centers for each topic

topic	before	after	topic	before	after	topic	before	after
Museum	66	98	Holiday	25	23	Flowers	41	26
Book	76	49	Pet	59	59	Luxury	33	33
Event	39	39	Beauty	42	42	Shop	98	101
Music	98	42	Car	98	98	Toy	41	41
Election	40	18	Chat	80	73	Developper	98	101
Finance	98	67	Dating	80	80	Hardware	98	98
Job	98	55	Spirit	34	30	Internet	98	42
Loan	31	31	Travel	98	54	Macintosh	37	7
College	98	101	Magazine	98	101	Unix	43	43
Kid	98	98	Newspaper	98	98	Wireless	80	80
Adult	98	98	Health	98	101	Windows	98	98
Gambling	87	87	Sport	98	91			
Movie	98	98	Winter	68	55	average	72.3	65.1
Game	98	98	Athletic	45	45			
PS2	36	36	Auction	38	37			
Family	74	74	Clothing	48	48			
Food	98	98	Electronics	48	48			
Gardening	98	98	Entertainment	48	23			

Besides these topics that our system performs well, there are some other topics that the system outputs rather unexpected results. For example, the inputs and outputs for topic Magazine is as follows:

- Inputs: chemweek.com, mysterynet.com, cosmomag.com, playbill.com, si.edu
- Outputs: washingtonpost.com, nytimes.com, usatoday.com, latimes.com, wsj.com

This result shows that the topic of the centers are shifted from Magazine to Newspaper, and it also shows the closeness of the communities of these two topics. Another example is the community for topic Travel:

- Inputs: smarterliving.com, sheraton.com, ebookers.com, qixo.com, hotel.com
- Outputs: hilton.com, hyatt.com, sheraton.com, marriott.com, holiday-inn.com

In general, when a target topic is too broad that contains many subtopics, there are several representative pages for the topic. In this example, since many hotel sites are included in the input URLs, the topic of the community is focused to hotels in the process of purification.

Both HITS and our method are based on the graph structures that are extracted locally from the vast Web network. Since our method acquires Web data during the process of purification, and renews the members of communities iteratively, it is expected that the method performs well even when the members of initial communities are not representative ones.

5 Concluding Remarks

This paper proposes a new method for purifying Web communities based on the graph structure of hyperlinks. Results of the system that is developed based on our method are also shown. The following points should be mentioned for "purifying" our future research targets:

- The method proposed in this paper is considered to be a method for searching a complete bipartite subgraph included in a graph that correspond to a community. Although the effectiveness of our method depends heavily on the graph structure of target communities, typical graph structure of Web communities is not clear. We have to study more about the model for such structures that fits well for actual Web communities.
- There is no standard test data set for evaluating systems for Web mining. The above experimentation is made on the assumption that URLs listed in each topic of 100hot.com are ranked in the order of relevance to the topic. However, this assumption is not always true. In the ranking used for our experimentation, the top-ranking URL for topic car is Microsoft.com !! In order to evaluate the performance of our system objectively, some kind of standard test data set for Web mining is really needed.

References

1. K. Bharat, M. Henzinger: "Improved Algorithms for Topic Distillation in a Hyperlinked Environment", Proc. of the 21st Int'l ACM SIGIR Conf. pp.104-111, 1998.
2. A. Broder et. al.: "Graph structure in the Web", Proc. of WWW9 conference, 2000.
3. S. Chakrabarti et. al.: "Experiments in Topic Distillation", Proc. of ACM SIGIR workshop on Hypertext Information Retrieval on the Web, 1998.
4. S. Chakrabarti, et. al.: "Mining the Web's Link Structure", IEEE Computer, Vol.32, No.8, pp.60-67, 1999.
5. D. Gibson, J. Kleinberg, P. Raghavan: "Inferring Web Communities from Link Topology", Proc. of the 9th Conf. on Hypertext and Hypermedia, 1998.
6. M. Henzinger: "Hyperlink Analysis for the Web", IEEE Internet Computing, Vol.5, No.1, pp.45-50, 2001.
7. J. Kleinberg et. al.: "The Web as a Graph: Measurements, Models, and Methods", Proc. of COCOON '99, LNCS 1627, pp.1-17, Springer, 1999.
8. R. Kosala, H. Blockeel, "Web Mining Research: A Survey", ACM SIGKDD Explorations, Vol.2, No.1, pp.1-15, 2000.
9. R. Kumar et. al.: "Trawling the Web for Emerging Cyber-Communities", Proc. of the 8th WWW conference, 1999.
10. T. Murata: "Machine Discovery Based on the Co-occurrence of References in a Search Engine", Proc. of DS99, LNAI 1721, pp.220-229, Springer, 1999.

11. T. Murata: “Discovery of Web Communities Based on the Co-occurrence of References”, Proc. of DS2000, LNAI 1967, pp.65-75, Springer, 2000.
12. L. Page et. al.: "The PageRank Citation Ranking: Bringing Order to the Web", Online manuscript, http://www-db.stanford.edu/~backrub/pageranksub.ps, 1998.

Divide and Conquer Machine Learning for a Genomics Analogy Problem (Progress Report)

Ming Ouyang[1], John Case[2], and Joan Burnside[3]

[1] Environmental and Occupational Health Sciences Institute
UMDNJ – Robert Wood Johnson Medical School and
Rutgers, The State University of New Jersey
Piscataway, NJ 08854 USA
ouyang@fidelio.rutgers.edu
[2] Department of CIS
University of Delaware
Newark, DE 19716 USA
case@cis.udel.edu
[3] Department of Animal & Food Sciences
University of Delaware
Newark, DE 19716 USA
joan@udel.edu

Abstract. Genomic strings are not of fixed length, but provide one-dimensional spatial data that do *not* divide for conquering by machine learning into manageable fixed size chunks obeying Dietterich's independent and identically distributed assumption. We nonetheless need to divide genomic strings for conquering by machine learning — in this case for genomic prediction.

Orthologs are genomic strings derived from a common ancestor and having the same biological function. Ortholog detection is biologically interesting since it informs us about protein divergence through evolution, and, in the present context, also has important agricultural applications. In the present paper is indicated means to obtain an associated (fixed size) attribute vector for genomic string data and for dividing and conquering the machine learning problem of ortholog detection herein seen as an analogy problem. The attributes are based on both the typical string similarity measures of bioinformatics and on a large number of differential metrics, many new to bioinformatics. Many of the differential metrics are based on evolutionary considerations, both theoretical and empirically observed, in some cases observed by the authors.

C5.0 with AdaBoosting activated was employed and the preliminary results reported herein re *complete* cDNA strings are very encouraging for eventually and usefully employing the techniques described for ortholog detection on the more readily available EST (incomplete) genomic data.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 290–303, 2001.

1 Introduction

Genomic strings are strings of one of two types: nucleotide strings and amino acid strings. Nucleotide strings are what genes are, and they code for amino acid strings which are proteins. We can model each as strings of letters where the letters are standard names for the nucleotides or the amino acids. For machine learning[1] purposes, it is not practical to process genomic strings as fixed-size vectors (of letters). However, genomic strings can be thought of as *one*-dimensional spatial structures.[2] Dietterich [Die00] discusses in detail the problem for machine learning of employing divide and conquer on spatial and temporal data which can't be practically *completely* represented as fixed-size vectors. Of course such data can be divided into manageable fixed size chunks. He notes, though, that divide and conquer is problematic if the data fails to satisfy the *independent and identically distributed* (*iid*) assumption. As we will see below, the problem discussed in this paper does not satisfy this assumption, and this paper provides, then, among other things, a case study of how in our problem domain we circumvent the difficulty.

In GenBank (major repository of genomic information) there are many human and mouse (mammal) genomic sequences with *known* associated functions; there are *some but fewer* (food animal) chicken sequences with known associated functions. Poultry is the third largest agricultural commodity, and the main meat consumed in the U.S.[3] Control of disease in these birds is important for both agricultural economics and human health. The identification of candidate genes for disease resistance, or the development of immune enhancers to make vaccines more effective or even obsolete are among the more contemporary approaches to disease control in this important food animal. However, gene sequence information for birds is currently too limited. Fortunately, as just noted above, *some* information *is* available, so there is some basis for training a machine learning procedure.

Orthologs are (genomic) sequences which are from different species but which have common descent and the same function. Crucially, in a number of cases one *can* locate and compare human, mouse, *and chicken* orthologs. We've been concerned, then, with an *analogy problem*: find/exploit *patterns* in the *known* orthologs between human, mouse, and chicken and apply those patterns to human and mouse orthologs X, Y with known function, but whose chicken ortholog Z is *un*known, to detect the unknown Z.

To find patterns between relatively closely related species, e.g., human and mouse, it has sufficed to use known local-alignment-based similarity tools such as BLAST (and variants) [AGM$^+$90,AMS$^+$97,KA90,Pea95] which are based on string *similarity* only. They find "locally maximal segment pairs." This similarity

[1] *Machine learning* [Mit97,RN95] involves algorithmic techniques for fitting programs to data and for outputting the programs fit for subsequent use in predicting future data. A program so fit to data is said to be *learned*.

[2] Amino acid sequences *fold* into 3-D structures, but that, for us, will be taken into account in future work. See Section 6 below.

[3] http://www.usda.gov/news/pubs/fbook98/ch1a.htm

matching does *not* suffice for highly *divergent* orthologs (e.g., some of the orthologs between mammals and birds) since the regions of similarity are too fragmented. For example, Figure 1 depicts an optimal global *amino acid* sequence alignment between chicken and mouse IL-2 orthologs[4] (with chicken shown on top). The corresponding *nucleotide* sequence alignment is also very fragmented (data not shown). The same degree of fragmentation is seen comparing chicken and human IL-2 (data not shown). When searching chicken IL-2 against GenBank, BLAST and variants do not and cannot find any locally maximal segment pairs in mammals which have statistical significance. This problem is not just for IL-2. More generally, it follows from [RYW+00] and recent news releases from Celera that more than 25% of orthologs are not identified by commonly used (local-alignment-based similarity) tools.

```
---------------MMCKVLIFGCISVATLMTTAYGASLSSAKRKPLQTLIKDL-EIL------ENIKNKIH
                                                |  |  || | |       | |
MYSMQLASCVTLTLVLLVNSAPTSSSTSSSTAEAQQQQQQQQQQQQHLEQLLMDLQELLSRMENYRNLKLPRM

LEL--YTPTETQECTQQTLQCY------LGEVVTLKKETEDDTEIKEEFVTAIQNIEKNLKSLTGLNHTGSEC
|    | |    |     |||       |  |  |        |  | |   |      ||   |      ||
LTFKFYLPKQATE--LKDLQCLEDELGPLRHVLDLTQSKSFQLEDAENFISNIRVTVVKLK---G-SDNTFEC

KICEANNKKKFPDFLHELTNFVRYLQK----
            |||     |
QF--DDESATVVDFLRRWIAFCQSIISTSPQ
```

Fig. 1. Optimal Global Amino Acid Alignment Between Chicken and Mouse IL-2

In the analysis of analogy problems from both cognitive psychology [Ste88] and artificial intelligence [Eva68,RN95], we see that both similarities *and differences* need to be taken into account. For *example*, here are a couple of string analogy problems from Hofstadter. These problems are based on alphabetical order, though, not genomics.

abc → abd, ijk → ?
abc → abg, iijjkk → ?

We see that taking into account both string similarities and differences are a necessary part of solving these problems.

Other projects have employed *differential* metrics to some degree and to good effect. The tools for intron-exon[5] recognition (not what we are doing in the present study), GRAIL [GME+92] and GENSCAN [BK97], employ differential metrics (and there is a similarity metric implicit, for example, in the potential function in GRAIL). A *codon* is comprised of a contiguous triple of nucleotides

[4] IL-2 is interleukin 2, an immune system protein.
[5] *Exons* contain the coding portions of genes.

```
chick-human AA identity <= 25.54: no
chick-human AA identity > 25.54:
:...chick-mouse NA identity <= 49.5:
    :...chick-human NA length/(# gaps) > 57.45: no
    :   chick-human NA length/(# gaps) <= 57.45:
    :   :...mouse A to chick C <= 19.09: no
    :       mouse A to chick C > 19.09: yes
    chick-mouse NA identity > 49.5:
    :...chick-human NA length/(# gaps) <= 118.0588:
        :...chick-mouse NA length/(# gaps) > 103.7143: no
        :   chick-mouse NA length/(# gaps) <= 103.7143:
        :   :...chick T to mouse C > 25.59: no
        :       chick T to mouse C <= 25.59:
        :       :...chick T to human G <= 13.39: yes
        :           chick T to human G > 13.39: no
        chick-human NA length/(# gaps) > 118.0588: [Rest omitted]
```

Fig. 2. First Tree Output By C5.0 — With Portion Omitted

{A, C, G, T}, and 61 of these triples each code for a single corresponding amino acid. Differential metrics can be based on so-called *codon bias* [SM82,SCH+88, Li97]. Most of the 20 amino acids are encoded by more than one codon; *codon bias* is, then, the quantifiable phenomenon that an organism uses one particular codon for an amino acid significantly more often than all the other synonymous codons. [SG94] provides an improvement of BLAST with a measure of codon bias as a differential metric. In the present project we employ codon bias as one class of differential metrics or attributes: we count, for each of the 61 codons, how many times it occurs in the orthologs.

In our project for (chicken) ortholog detection, we have devised a number of other differential metrics also to complement standard similarity metrics for genomic sequences. These measures of similarity and differences provide our attributes (or features) for machine learning and constitute, in many cases, a useful division into parts of the original problem about 1-D strings, a division towards conquering the problem. As noted above, this division yields cases where the *iid* assumption fails. Instead, the co-evolution of mammal and bird orthologs from common ancestor strings involves whole *interdependent* string patterns coming out partly differently and partly similarly.

2 Attributes Based on Similarities and Differences

We mentioned codon bias for differential metrics above.

A straightforward evaluator of similarity is simple percent identity. Studies ([Li97], Chapter 1) have shown *with accompanying simple biochemical explanation* that, when mutations occur, the nucleotides A and G tend to change to G and A, respectively, and C and T tend to change to T and C, respectively;

these are called *transitions*. The other 8 substitutions, between the group of $\{\mathrm{A}, \mathrm{G}\}$ and the group of $\{\mathrm{C}, \mathrm{T}\}$ are called *transversions*, and they occur less frequently than transitions. Insertions and deletions of nucleotides are thought to occur rarely; however, when they do occur, several adjacent nucleotides may be involved [BO98]. Therefore, another common way to evaluate the quality of an alignment is to assign a high score to identity matches, a medium score to transitions, a low score to transversions, a large penalty to opening a gap, and a small penalty to extending a gap. Our Table 1 is such a scoring scheme.

Table 1. A Nucleotide Sequence Alignment Scoring Scheme.

From\To	A	C	G	T
A	4	1	2	1
C	1	4	1	2
G	2	1	4	1
T	1	2	1	4

Gap Opening	Gap Extension
-5	-2

Amino acid sequences are what cells translate nucleotide (gene) sequences into (to form proteins). When amino acid sequences are aligned, the scoring matrix is a 20 by 20 table because there are 20 amino acids; some commonly used matrices for amino acid sequence alignment include PAM and BLOSUM families of matrices (see [BO98] and the references therein).

The Needleman-Wunsch algorithm [NW70] finds an optimal *global* alignment of two sequences. Optimal global alignment has thus far been mostly used in comprehensive studies of orthologs, as in [MB98], where orthology has already been established, and researchers want to extract additional information from the aligned sequences. Global alignment involves some increased complexity costs over local alignment schemes, but we've seen, for our applications reported herein, that this increased cost is not prohibitive; furthermore, we have begun using the more efficient variant of Needleman-Wunsch from [Got82]. When we apply (the improved variant of) Needleman-Wunsch to obtain global alignment values for similarity attributes, for nucleotide alignment we apply the scoring scheme in Table 1, and, for amino acid alignment, we apply the scoring scheme from the PAM250 matrix. Needleman-Wunsch and its improvement calculate a global alignment optimal in the sense that no other alignment yields a higher score, global in the sense that the entire lengths of the sequences are taken into consideration.

For our similarity attributes we employ both the Nucleotide Alignment (NA) scores and the Amino acid Alignment (AA) scores — comparing chicken with each of mouse and human.[6] These scores are given as percent identities.

[6] Applying attribute values for both chicken-mouse and chicken-human comparisons improves performance over just employing comparisons between chicken and one of these mammals.

We noticed an intriguing and biologically significant empirical pattern comparing NA and AA for our current full data set of 213 complete orthologs between chicken-mouse-human. In Table 2 this pattern (among other things) is displayed for a *representative* sample of 20 of our orthologs. Table 2 is shown *sorted in the column of chicken-mouse nucleotide alignment*. For the top portion of the table (as sorted), chicken-mouse NA percent identity is larger than that of AA, but for the bottom portion of the table, the ordering of the two numbers becomes reversed. The likely biological/biochemical explanation appears to be: in the top portion we see the effects of mutations in third and redundant position in codons [Li97]; in the bottom portion we see critically preserved amino acids; and in the middle some combination of each. We have employed, then, the values of (NA-AA) and NA/AA as attributes measuring the degree to which nucleotide and amino acid alignments differ.

From the NA and AA alignments themselves we calculate lengths, numbers of gaps, and their average lengths. We then combine these numbers importantly in various *ratios* to provide differential attributes. The present paper reports on our progress with ortholog prediction for *complete* cDNA sequences. In the future we plan to apply our methods to ESTs (incomplete sequence data), and, making these attributes ratios is one way that, on average, the incompleteness of the ESTs will not bias our attributes compared with their values on complete sequences.

We also employ as attributes the percentages of conservations of the four nucleotides, the percentages of transitions, and the percentages of transversions.

From above, *transversions* are those nucleotide mutations (e.g., from C or T to A) that are less likely to occur biochemically. Table 2 also displays, for the 20 representative orthologs (out of our 213) *transversion bias* percents between mouse and chicken. We've based a number of additional attributes on various measures suggested by the biologically quite interesting transversion bias trends seen in this table *and in the table of all our 213 orthologs*. E.g., we have various useful attributes measuring deviation from the boldfaced columns for transversion bias.

We illustrate these attributes with the example of a particular chicken sequence compared to its mouse ortholog. For such a sequence comparison (corresponding to the first four columns of a single row of a table like Table 2) there are four transversion percentages: $(t_1, t_2, t_3, t_4) =$

$$(\%\text{ of }\{\text{C,T}\} \to \text{A}, \%\text{ of }\{\text{A,G}\} \to \text{C}, \%\text{ of }\{\text{C,T}\} \to \text{G}, \%\text{ of }\{\text{A,G}\} \to \text{T}). \quad (1)$$

We treat these four numbers as coordinates of a four-dimensional point. The *general* pattern (quite similar to that of the boldface pattern in Table 2): for transversions from chicken to mouse, a point will usually have a larger/largest second-coordinate than its other coordinates; hence, the points will reside in a restricted sub-region in the space. Since the distribution of these points is not known, we could use the distribution-free, scaling and rotation invariant measure called *simplicial depth* [BF84,Liu90,LS93,CO01] to measure how near a point is to the center of the cluster of points. We have experimented, to good effect

Table 2. Transversion Bias and Comparative Aligments

Protein	Chicken to mouse				Mouse to chicken					
From	CT	AG	CT	AG	CT	AG	CT	AG	% identity	
To	A	C	G	T	A	C	G	T	NA	AA
frizzled 7	2.2	**8.2**	6.9	2.9	1.7	**8.6**	7.7	2.0	81.8	87.4
transforming growth factor β 3	**6.0**	5.7	4.0	1.9	2.9	**6.7**	5.3	2.6	81.1	87.1
nicotinic Ach receptor α 1	3.1	**6.6**	6.1	2.5	5.1	**6.5**	3.2	3.5	79.4	84.3
growth hormone	4.0	**9.1**	6.4	5.0	5.0	7.1	**8.3**	3.9	74.8	73.1
VEGF	5.9	**8.2**	3.5	1.6	5.7	3.9	**6.1**	3.9	74.7	73.4
PDGF receptor α	4.7	**7.0**	5.8	3.2	**7.0**	3.9	4.9	5.2	74.3	79.3
estrogen receptor	3.2	**9.7**	6.1	3.3	**8.7**	4.2	4.7	4.8	73.9	78.3
PDGF α	5.7	**8.2**	6.1	2.4	**7.9**	4.0	5.2	5.6	72.2	76.7
FSH receptor	4.9	**7.9**	4.9	6.1	**8.1**	5.8	4.4	5.2	71.5	71.6
fibroblast growth factor 2	4.9	**9.7**	8.8	5.5	8.0	5.3	**9.0**	7.0	70.0	66.0
thyrotropin β	5.2	**8.2**	6.2	8.2	7.8	4.8	6.9	**8.0**	69.8	65.4
growth hormone receptor	**9.2**	7.0	5.4	7.1	**9.8**	6.0	6.0	7.0	66.6	56.9
insulin-like growth factor I	6.6	**11.4**	6.6	5.0	**11.5**	6.2	5.8	6.2	64.1	62.9
prolactin	9.5	**9.6**	7.2	8.7	**10.0**	8.1	7.4	9.3	62.2	50.8
β 2 microglobulin	17.8	**18.3**	11.7	11.7	7.7	**22.9**	22.1	6.7	54.7	42.9
prolactin receptor	8.8	**10.1**	6.7	8.5	**14.1**	6.6	7.7	6.5	54.6	42.8
interleukin 1 β	**18.8**	11.6	11.2	13.2	11.6	**21.0**	14.2	7.8	51.3	31.7
interleukin 18	**14.6**	13.3	11.7	11.3	**15.1**	10.4	14.3	11.4	51.2	31.8
interleukin 15	10.8	**14.8**	8.8	13.7	**23.3**	6.6	9.8	9.9	49.6	33.8
interleukin 2	**19.3**	11.7	13.9	16.7	**24.9**	9.0	10.4	17.5	42.5	19.9

Left: Transversion bias %s
(the largest number in each row is **boldfaced**).

Right: Nucleotide (NA) vs. Amino acid (AA) sequence Alignments as % identities.
Rows sorted by NA.

(Section 5), with easily computed, one-dimensional projections of the full, more difficult to compute simplicial depth: we see not only that t_2 tends to be the largest of the four, but, when t_2 is not the largest, that t_1 tends to be. We use as one-dimensional projections, the following formulas for additional differential attributes: $t_2/\text{minimum}(t_1, t_2, t_3, t_4)$ & $t_1/\text{minimum}(t_1, t_2, t_3, t_4)$. The first we call a *major transversion bias*, the second a *minor*. Similar (but not the same) formulas are used for the transversion biases from mouse to chicken, and for those between human and chicken. Relatively large values in these differential measures indicate conformation to the typical transversion bias patterns.

Lastly we employ some simple protein class information [AKF+95][7] (see also [TSB00]) for attributes.

[7] http://www.tigr.org/docs/tigr-scripts/egad_scripts/role_report.spl

3 How We Obtain Negative Data for Classification

For the classification of genomic sequences as orthologous or not we want to supply for training data both positive *and negative* instances.

Our positive data come from our 213 known orthologs.

We employ two groups of negative data. The first group is of the form

$$(\text{ human protein } Y,\ \text{mouse protein } Y,\ \text{chicken protein } X\ ), \tag{2}$$

where

- X and Y are in our collection,
- X and Y are not orthologs, and
- the two differences in lengths between chicken and each mammal protein is less than 30% of the length of the mammalian protein, and at least one of the amino acid global alignment identities between chick X and human Y or between chick X and mouse Y is greater than or equal to 13% (The 30% and 13% figures may be adjusted in the future as appears necessary, etc).

For our 213 orthologs, there are 1043 data points in the first group. This first group corresponds to the type of negative data points on which we would want to test a decision program output by a machine learning technique. The second group is of the two forms

$$(\text{ human protein } X,\ \text{mouse protein } Y,\ \text{chicken protein } X\ ) \tag{3}$$

and

$$(\text{ human protein } Y,\ \text{mouse protein } X,\ \text{chicken protein } X\ ), \tag{4}$$

and the constraints on the proteins are the same as in the first group. For the 213 orthologs, there are 2086 data points in the second group. The use of this second group considerably improves performance of decision programs output by the machine learning technique described in the next section.

4 Machine Learning Techniques Employed

We employ as our machine learning technique Quinlan's C5.0 which combines his C4.5 for decision tree induction [Qui93,RN95,Mit97] with the option for AdaBoosting [FS97,FS99]. *Decision tree induction* involves the fitting of simple decision trees with unary-predicate tests to classification data. C5.0 (and C4.5) employ an information-theoretic heuristic so that *decisions at the top of a tree fitted explain more data than decisions further down.* This provides both *efficiency* in fitting and some *readability of the resultant trees for insight* — the reasons the decision tree induction component was chosen for the project. AdaBoost is an important technique for improving learners both for fitting training data [FS96] *and for generalization and prediction* beyond the training data [FSBL98] (see also [FMS01]). It also handles well the presence of errors in the training data

[FS96]. AdaBoosting, as employed in C5.0, takes a weighted majority vote of the decisions of a *sequence of* decision trees, where each tree, beyond the first, judiciously concentrates on the cases difficult for its predecessor.[8] Since AdaBoosting combines a number of decision trees, its use may involve some tolerable loss of readability and efficiency. However, AdaBoosting nonetheless looks like *linear* (i.e., fast) programming [FS99]. The features of AdaBoosting just mentioned are why it was chosen for the project.

Other methods might have been chosen. Reported in [MST94] is a major series of studies and domains comparing machine learning techniques (including decision tree induction and neural net learning) and classical statistical techniques. Decision tree induction was generally robust over the domains studied including compared to statistics. Again, though, it had the advantage that it's products are readable for insight. Of course, each technique compared had its especially good domains. In [BB98], for example, we see many bioinformatics problems tackled with either neural nets or statistical techniques (but not decision tree induction with AdaBoosting). Neural nets and statistical methods tend not to produce classification programs readable for insight. We do note that the Morgan system in [SDFH98] does employ decision tree induction — to simplify otherwise complex dynamic programming for doing similarity matching for intron-exon recognition in vertebrates.[9] We also see that [AMS$^+$93] employs a decision tree induction which automatically selects string patterns from a given table and produces a decision program which tests input data against the table to predict transmembrane domains from protein data. Support vector machines [Vap95,Vap98] and neural nets can, in effect, cut up the attribute space in ways that decision trees do not. For example, in some cases, there can be advantages in decision tree induction to suitably rotate the attribute space; *however*, AdaBoosting (more than) makes up for any such advantage [Qui97], and, in effect, cuts up the attribute space very finely [Qui98]. Furthermore, support vector machines involve *quadratic* (i.e., slower) programming [FS99].

5 Results

When we run C5.0 with AdaBoost activated on our 213 orthologs (and associated negative data) we get ensembles of decision trees with an average of about 35 decision nodes per tree. These trees are humanly readable. The attributes tested in ensembles of trees based all 213 orthologs involve most of our current attributes. The decisions made by such an ensemble with only three trees[10] makes *no* mistakes on all the positive and negative data points generated by the

[8] Importantly, the voting weights are bigger for more accurate trees in the sequence of trees.

[9] In the present project we are working only with exons or portions thereof.

[10] Recall from Section 4 above that the ensemble of trees obtained from AdaBoosting makes its decisions by a judiciously weighted majority vote among the decisions of its constituent trees — even more usefully subtle decision making than that of any single tree.

213 orthologs. More importantly, though, we employed 10-fold cross-validation (i.e., a random 10-th of the data is removed from training and employed instead for testing) with 10 repetitions and obtained, with an boosting ensemble size of 25 trees, a low error rate of 2.4% (with Standard Error less than 0.05%) on the entire data set for all 213 orthologs. Furthermore, for each of the 213 ways of removing one ortholog of the 213, we also tried training on the remaining 212 (with their associated negative data points) and testing the ensembles obtained from C5.0 with AdaBoost activated on the missing ortholog and the (also missing) negative data points associated with it. In 95% of the cases that the ortholog omitted from the training data was chosen from the important protein class of *cell/organism defense* (which includes the immune system enhancers we are especially interested in)[11], ensembles with only four trees performed perfectly on all the positive and negative cases including those for the ortholog omitted.

On our 213 orthologs and associated negative data the first decision tree produced by C5.0 (with portion omitted to save space) is shown in Figure 2. The tree should be read essentially as an **if-then-else** program with nesting indicated by indentation. The decision `yes` is for othologous and `no` is for non-othologous. From vertical position in the tree we see, for *example*, that the top test of an amino acid percent identity, `chick-human AA identity <= 25.54`, explains more data than the test somewhat below of a transversion percent, `chick T to human G <= 13.39`. In the omitted portion there appears, among other tests, the test `chick-mouse transversion bias (minor) > 2.292322`

A conclusion of all these results is that, at least for complete cDNA sequences, with only our current attribute set, we can apparently explain or cover most of the causes behind orthology between the chosen bird and mammal species. Our paper also presents a particularly successful application of the machine learning method chosen (C5.0).

6 Future Work

In the future we need to expand our search for detectable but currently unknown chicken orthologs in large and rich databases of chicken ESTs. GenBank and, importantly, our own expanding database of now over 17,000 expressed chicken ESTs[12] will be crucial sources. Based on the results of the previous section we are quite optimistic that, when we train on both complete and EST data in the interest of ortholog detection also for ESTs, we can be successful.

A number of further enhancements of the machine learning techniques are also planned, including learning motif (i.e., pattern) information (as in [AMS+93]) and employment of further machine learning modules such as FOCL [PBS91,Mit97] to enable better use of *explicit* background information from empirical and theoretical evolutionary considerations (such background information is implicit in some of the above). *Multi-tasking* is learning more than one thing

[11] http://www.tigr.org/docs/tigr-scripts/egad_scripts/role_report.spl
[12] http://www.chickest.udel.edu

for the purpose of mutual enhancement of the learning. It is shown helpful empirically [SR86,PMK91,Car93,MCF+94,DHB95,Car96,MK96,TS96,BGN97] and theoretically, [Ash60,AGS89,KSVW93,KS94,CJO+00]. In the future we hope to enhance our ortholog prediction by multi-tasking it with also *learning* protein classes. We also plan to try as additional attributes global alignment metrics on the strings coding folding structure that are output by *nnpredict* [KCL90].[13] This algorithm produces good but approximate predictions [KCL90], so it will be interesting to see if its predictions help or hinder ours.

We expect the work preliminarily described herein will help significantly with speeding up the discovery of useful new orthologies and also provide general insights regarding the evolutionary divergence between distantly related species (e.g., bird and mammal species). We anticipate agriculturally important applications, e.g., it may lead to a reduction in the use of antibiotics in poultry. We also plan to apply our techniques to other species, e.g., zebrafish, fugu and frog.

Acknowledgement. Ming Ouyang was partially supported by a postdoctoral fellowship of Delaware Biotechnology Institute and by the USEPA funded Center for Exposure and Risk Modeling (CR827033).

References

[AGM+90] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.

[AGS89] D. Angluin, W. Gasarch, and C. Smith. Training sequences. *Theoretical Computer Science*, 66(3):255–272, 1989.

[AKF+95] M.D. Adams, A.R. Kerlavage, R.D. Fleischmann, R.A. Fuldner, C.J. Bult, N.H. Lee, E.F. Kirkness, K.G. Weinstock, J.D. Gocayne, O. White, and et al. Initial assessment of human gene diversity and expression patterns based upon 83 million nucleotides of cDNA sequence. *Nature*, 377:3–174, 1995.

[AMS+93] S. Arikawa, S. Miyano, A. Shinohara, S. Kuhara, Y. Mukouchi, and T. Shinohara. A machine discovery from amino-acid-sequences by decision trees over regular patterns. *New Generation Computing*, 11:361–375, 1993.

[AMS+97] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[Ash60] R. Ashby. *Design for a Brain: The Origin of Adaptive Behavior.* Wiley, NY, second edition, 1960.

[BB98] P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach.* MIT Press, Cambridge, MA, third edition, 1998.

[BF84] E. Boros and Z. Füredi. Triangles covering the centre of an n-set. *Geometriae Dedicata*, 17:69–77, 1984.

[13] http://www.cmpharm.ucsf.edu/∼nomi/nnpredict.html & http://www.cmpharm.ucsf.edu/∼nomi/nnpredict-instrucs.html

[BGN97] Kai Bartlmae, Steffen Gutjahr, and Gholamreza Nakhaeizadeh. Incorporating prior knowledge about financial markets through neural multitask learning. In *Proceedings of the Fifth International Conference on Neural Networks in the Capital Markets*, 1997.

[BK97] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, 268:78–94, 1997.

[BO98] Andreas D. Baxevanis and B.F. Francis Ouellette, editors. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. John Wiley & Sons, Inc., 1998.

[Car93] Richard A. Caruana. Multitask connectionist learning. In *Proceedings of the 1993 Connectionist Models Summer School*, pages 372–379, 1993.

[Car96] R. Caruana. Algorithms and applications for multitask learning. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML-96)*, pages 87–95. Morgan Kaufmann, San Francisco, CA, 1996.

[CJO+00] J. Case, S. Jain, M. Ott, A. Sharma, and F. Stephan. Robust learning aided by context. *Journal of Computer and System Sciences (Special Issue for* COLT'98*)*, 60:234–257, 2000.

[CO01] Andrew Y. Cheng and Ming Ouyang. On algorithms for simplicial depth. In *13th Canadian Conference on Computational Geometry*, pages 53–56. University of Waterloo, August 13-15 2001.

[DHB95] Thomas G. Dietterich, Hermann Hild, and Ghulum Bakiri. A comparison of ID3 and backpropogation for English text-to-speech mapping. *Machine Learning*, 18(1):51–80, 1995.

[Die00] T. Dietterich. The divide-and-conquer manifesto. In *Proceedings of The 11th International Workshop on Algorithmic Learning Theory (ALT'00)*, Lecture Notes in Artificial Intelligence, pages 13–26. Springer-Verlag, Berlin, 2000.

[Eva68] T. Evans. A program for the solution of a class of geometric-analogy intelligence-test questions. In M. Minsky, editor, *Semantic Information Processing*, pages 271–353. MIT Press, 1968.

[FMS01] Y. Freund, Y. Mansour, and R. Schapire. Why averaging classifiers can protect against overfitting. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, 2001.

[FS96] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML-96)*, pages 148–156. Morgan Kaufmann, San Francisco, CA, 1996.

[FS97] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[FS99] Y. Freund and R. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999. In Japanese and translated by Naoki Abe; English version at http://www.research.att.com/∼schapire/cgi-bin/uncompress-papers/FreundSc99.ps.

[FSBL98] Y. Freund, R. Schapire, P. Bartlett, and W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[GME+92] X. Guan, R.J. Mural, J.R. Einstein, R.C. Mann, and E.C. Uberbacher. GRAIL: An integrated artificial intelligence system for gene recognition and interpretation. In *Eighth IEEE Conference on AI Applications*, pages 9–13, Monterey, CA, March 2-6 1992. IEEE Computer Society Press.

[Got82] O. Gotoh. An improved algorithm for matching biological sequences. *J. Mol. Biol.*, 162:705–708, 1982.

[KA90] Samuel Karlin and Stephen F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. USA*, 87:2264–2268, 1990.

[KCL90] D. G. Kneller, F. E. Cohen, and R. Langridge. Improvements in protein secondary structure prediction by an enhanced neural network. *Journal of Molecular Biology*, 214:171–182, 1990.

[KS94] M. Kummer and F. Stephan. Inclusion problems in parallel learning and games. In *Proceedings of the Workshop on Computational Learning Theory*, pages 287–298. ACM Press, NY, July 1994. Journal version to appear, *Journal of Computer and System Sciences (Special Issue for* COLT'94*)*, 52(3):403-420, 1996.

[KSVW93] E. Kinber, C. Smith, M. Velauthapillai, and R. Wiehagen. On learning learning multiple concepts in parallel. In *Proceedings of the Workshop on Computational Learning Theory*, pages 175–181. ACM, NY, 1993.

[Li97] Wen-Hsiung Li. *Molecular Evolution.* Sinauer Associates, Inc., 1997.

[Liu90] R.Y. Liu. On a notion of data depth based on random simplices. *The Annals of Statistics*, pages 405–414, 1990.

[LS93] R.Y. Liu and K. Singh. A quality index based on data depth and multivariate rank tests. *Journal of American Statistical Association*, 88:252–260, 1993.

[MB98] Wojciech Makalowski and Mark S. Boguski. Evolutionary parameters of the transcribed mammalian genome: An analysis of 2,820 orthologous rodent and human sequences. *Proc. Natl. Acad. Sci. USA*, 95:9407–9412, 1998.

[MCF+94] T. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski. Experience with a learning, personal assistant. *Communications of the ACM*, 37:80–91, 1994.

[Mit97] T. Mitchell. *Machine Learning.* McGraw Hill, 1997.

[MK96] S. Matwin and M. Kubat. The role of context in concept learning. In M. Kubat and G. Widmer, editors, *Proceedings of the ICML-96 Pre-Conference Workshop on Learning in Context-Sensitive Domains, Bari, Italy*, pages 1–5, 1996.

[MST94] D. Michie, D. Spiegelhalter, and C. Taylor, editors. *Machine Learning, Neural and Statistical Classification.* Ellis Horwood, NY, 1994.

[NW70] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48:443–453, 1970.

[PBS91] M.J. Pazzani, C.A. Brunk, and G. Silverstein. A knowledge-intensive approach to learning relational concepts. In L. Birnbaum and G. Collins, editors, *Proceedings of the 8th International Workshop on Machine Learning*, pages 432–436. Morgan Kaufmann, 1991.

[Pea95] William R. Pearson. Comparison of methods for searching protein sequence databases. *Protein Science*, 4:1145–1160, 1995.

[PMK91] L. Pratt, J. Mostow, and C. Kamm. Direct transfer of learned information among neural networks. In *Proceedings of the 9th National Conference on Artificial Intelligence (AAAI-91)*, 1991.

[Qui93] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[Qui97] J.R. Quinlan, 1997. Private communication.

[Qui98] R. Quinlan. Miniboosting decision trees. *Journal of AI Research*, 1998.

[RN95] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, NJ, 1995.

[RYW+00] Gerald M. Rubin, Mark D. Yandell, Jennifer R. Wortman, George L. Gabor Miklos, Catherine R. Nelson, Iswar K. Hariharan, Mark E. Fortini, Peter W. Li, Rolf Apweiler, Wolfgang Fleischmann, J. Michael Cherry, Steven Henikoff, Marain P. Skupski, Sima Misra, Michael Ashburner, Ewan Birney, Mark S. Boguski, Thomas Brody, Peter Brokstein, Susan E. Celniker, Stephen A. Chervitz, David Coates, Anibal Cravchik, Andrei Gabrielian, Richard F. Falle, William M. Gelbart, Reed A. George, Lawrence S.B. Goldstein, Fangcheng Gong, Ping Guan, Nomi L. Harris, Bruce A. Hay, Roger A. Hoskins, Jiayin Li, Zhenya Li, Richard O. Hynes, S.J.M. Jones, Peter M. Kuehl, Bruno Lemaitre, J. Troy Littleton, Debrah K. Morrison, Chris Mungall, Patrick H. O'Farrell, Oxana K. Pickeral, Chris Shue, Leslie B. Vosshall, Jiong Zhang, Qi Zhao, Xiangqun H. Zheng, Fei Zhong, Wenyan Zhong, Richard Gibbs, J. Craig Wenter, Mark D. Adams, and Suzanna Lewis. Comparative genomics of the eukaryotes. *Science*, 287:2204–2215, 2000.

[SCH+88] Paul M. Sharp, Elizabeth Cowe, Desmond G. Higgins, Denis C. Shields, Kenneth H. Wolfe, and Frank Wright. Codon usage patterns in *escherichia coli, bacillus subtilis, saccharomyces cerevisiae, schizosaccharomyces pombe, drosophila melanogaster* and *homo sapiens*: a review of the considerable within-species diversity. *Nucleic Acids Research*, 16(17):8207–8211, 1988.

[SDFH98] Steven Salzberg, Arthur L. Delcher, Kenneth H. Fasman, and John Henderson. A decision tree system for finding genes in DNA. *Journal of Computational Biology*, 5(4):667–680, 1998.

[SG94] David J. States and Warren Gish. Combined use of sequence similarity and codon bias for coding region identification. *Journal of Computational Biology*, 1(1):39–50, 1994.

[SM82] R. Staden and A.D. McLachlan. Codon preference and its use in identifying protein coding regions in long DNA sequences. *Nucleic Acids Research*, 10(1):141–156, 1982.

[SR86] Terrence J. Sejnowski and Charles Rosenberg. NETtalk: A parallel network that learns to read aloud. Technical Report JHU-EECS-86-01, Johns Hopkins University, 1986.

[Ste88] R. Sternberg. *The Triarchic Mind*. Viking, NY, 1988.

[TS96] S. Thrun and J. Sullivan. Discovering structure in multiple learning tasks: The TC algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML-96)*, pages 489–497. Morgan Kaufmann, San Francisco, CA, 1996.

[TSB00] V. Tirunagaru, L. Sofer, and J. Burnside. An expressed sequence tag database of activated chicken T cells: Sequence analysis of 5000 cDNA clones. *Genomics*, 2000. In press.

[Vap95] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

[Vap98] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, 1998.

Towards a Method of Searching a Diverse Theory Space for Scientific Discovery

Joseph Phillips

University of Pittsburgh
Computer Science Dept.
Pittsburgh, PA 15260, USA
josephp@cs.pitt.edu

Abstract. Scientists need *customizable* tools to help them with discovery. We present an adjustable heuristic function for scientific discovery. This function may be considered in either a Minimum Message Length (MML) or a Bayesian Net manner. The function is approximate because the default method of specifying theory prior probabilities is a gross estimate and because there is more to theory choice than maximizing probability. We do, however, effectively capture some user preferences with our technique. We show this for the qualitatively different domains of geophysics and sociology.

1 Introduction

Our ultimate goal is to write a general program to assist scientists in creating and improving scientific models. Realizing this goal requires progress in machine learning, knowledge discovery in databases, data visualization and search algorithms. It also requires progress in scientific model preferencing. The scientific model preference problem is compounded by the fact that several scientists with very similar background knowledge may see the same data but may prefer different models. This paper is the first in an on going study to address scientific model preferencing issue.

Scientific discovery can be viewed as a parameter search in a large and extremely inhomogeneous space. Physicists, for example, prefer strong relationships between numeric values (*e.g.*, equations) when they can be found. They also, however, use knowledge that is more conveniently expressed hierarchically in decision trees and semantic nets. This is exemplified by the classification of, and the assigning of fundamental properties to subatomic particles.

The minimum message length (MML) criterion is a mathematically well-grounded approach for choosing the most probable theory given data [21][8][24][5]. Inspired by information theory, the criterion states that the most probable model has the smallest encoding of both the theory and data. Ideally, the theory's encoding

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 304-322, 2001.

results from a domain expert's estimation of its prior probability and is language independent. The encoding of the data should also be probabilistic: as a function of a given theory.

Despite its generality and power for finding parameters in single classes of models (*e.g.*, the class of polynomials), many have expressed skepticism about whether MML may meaningfully be applied to finding parameters in inhomogeneous model spaces (*e.g.*, general scientific discovery). Cheeseman, for example, states "although finding the most probable domain model is often regarded as the goal of scientific investigation, in general, it is not the optimal means of making predictions." [5]

Our immediate, limited goal is to devise a heuristic function that can help users in large and inhomogeneous model spaces. Ideally, a search algorithm that is informed with our heuristic will return several regions in the model space that contain promising models, some known and some novel. Our approach is to adapt MML in a customizable manner.

1. We make MML applicable to a larger set of scientific discovery by mapping its terms onto those used by scientists: theory, laws and data. The MML theory is mapped to scientific theory. The MML data is split into scientific laws and data.
2. We make our heuristic function adjustable, but in a principled manner, by giving the user only two calibration parameters. These parameters directly correspond to the relationship between scientific theory and law, and scientific theory and data. It would be nice if we could ignore differences between theories and pretend that there is one "best" theory for all scientists. This, however, ignores significant evidence that scientists differ in opinion, *e.g.*, see [10][15].

We judge our function based on criteria for heuristic functions: generality, ease of computation, simplicity and smoothness.

We do *not* claim that we have "solved" this problem. The feature set by which to judge theories and the identification of the "best" model remain unsolved problems.

1. We offer no good guidance in developing the theory's prior probability. Cheeseman and others have stressed the importance of using domain knowledge to specify the theory's prior probability. They have also stated that syntactic features are often a poor substitute. We are aware of no general algorithm for the estimation of a theory's prior probability. Although our technique is not limited to syntactic features, we use them in this paper. Our approach is compatible with more principled prior probability specifying techniques.
2. We make no claim that the "best" theory will result from this approach. This is due to (1) the unsolved prior probability problem, (2) to the difficulty in searching a large and inhomogeneous model space, and (3) the fact that the most probable model may or may not be the best model.

We have developed a useful heuristic function despite these two major limitations. Its generality is tested by analyzing its performance in two completely different domains: sociology and geophysics.

This paper is organized as follows. Section 2 discusses previous approaches to automated scientific discovery. Section 3 briefly introduces MML. Our approach is detailed in section 4. Section 5 presents and discusses our experiments. Section 6 concludes.

2 Scientific Discovery

Several criteria have been proposed by philosophers of science for comparing competing hypotheses [3]. Among them are accuracy/empirical support, simplicity, novelty and cost/utility. Most automated approaches consider accuracy and simplicity.

IDS by Nordhausen and Langley was perhaps the first *general* program for scientific discovery [18][19]. IDS takes as input an initial hierarchy of abstracted states and a sequential list of "histories" (qualitative states, see [6]). Using each history IDS modifies the affected nodes of the abstracted state tree to incorporate any new knowledge gained from that history. Its output is a fuller, richer hierarchy of nodes representing history abstractions.

Thagard introduced Processes of Induction (or PI), to propose a computational scheme for scientific reasoning and discovery, but not as a working discovery tool [23]. PI represents models as having theories, laws and data. It evaluates scientific models by multiplying a simplicity metric by a data coverage metric. The simplicity metric is a function of how many facts have been explained and of how many co-hypotheses were needed to help explain them. The evaluation scheme is fixed and has no notion of degree of inaccuracy.

Zytkow and Zembowicz developed 49er, a general knowledge discovery tool [27][26]. It has a two stage process for finding regularities in databases. The first stage creates contingency tables (counts of how often values of one attribute co-occur with those of another) for pairings of database attributes. The second stage uses the contingency tables to constrain the search for other, higher order, regularities (*e.g.* taxonomies, equations, subset relations, *etc.*)

Valdes-Perez has suggested searching the space of scientific models from the simplest to ones with increasingly more complexity, stopping at the first that fits the data. MECHEM uses this approach to find chemical reaction mechanisms [25]. Such orderings would be easy to encode as heuristic functions.

We extend these approaches by using an adjustable, explicitly mentioned heuristic function that does not require enumerating all possible models. Our approach is to generalize Thagard's scheme and place it on sounder theoretical footing.

3 Information Theory and Diverse Model Discovery

The MML criterion is to minimize the sum of the length of a theory and data given the theory. Some data will have a smaller combined compressed length than the original message. For example, the pitch and relative durations of some bird calls may be written in musical notation. This notation dramatically reduces the information from the original time-dependent air-pressure signal that the bird produced. However, many sounds are not appropriately described by musical notation (*e.g.*, human speech). The original time-dependent air-pressure signal will be a better representation than musical notation.

The equation that relates these terms for data set D; context c; discrete, mutually exclusive and exhaustive hypotheses $\{H_0, H_1 .. H_n\}$ with assigned prior probabilities $p(H_i|c)$; and computed conditional data probabilities $p(D|H_i,c)$ is:

$$-\log p(H_i|D, c) = -\log p(H_i|c) - \log p(D|H_i, c) + const \quad (1)$$

which is equation (2) of [5]. Recall that the *-log(p(choice))* is the Shannon lower bound on the information needed to distinguish *choice* from other possibilities. The constant term serves to "normalize" the probabilities and may be ignored if you only want their relative order. Cheeseman gives this iterative process for applying MML:

1. Define the theory space.
2. Use domain knowledge to assign prior probabilities to the theories.
3. Use Bayes' theorem to obtain the posterior probabilities of the theories given the data from adequate descriptions of the theories (*i.e.*, from descriptions that let you compute $p(D|H_i,c)$).
4. Search the space with an appropriate algorithm.
5. Stop the search when a probable enough theory has been found (subject to computational constraints), or to redefine the theory space or prior probabilities.

Several obstacles hamper efforts to apply MML to general scientific discovery. Among them are the specification of the initial theory prior probabilities, the inherently iterative nature of MML, and the difficulty in searching this space for a true "highest probability" theory.

Like other MML efforts, there is no good rule for specifying an initial set of prior probabilities. Although Cheeseman and others warn about using syntactic features, this may be the easiest approach to try in a new domain.

MML is an inherently iterative process of redefining theory spaces and prior probabilities. This complicates the usage of any function that needs calibration.

The scientific theory search space is expected to be highly irregular, hampering the search for the "best" model. This is true of other domains. Cheeseman suggests simulated annealing and the EM algorithm as potential search mechanisms.

4 Our Approach

We do *not* claim to have an optimal heuristic function in terms of returning the truly "best" model. Rather, our goal is to create a decent heuristic function that may help scientists on their initial searches with large, inhomogeneous spaces.

Good heuristics for real-world problems are often tricky to design [16]. We evaluate our function based on four criteria:

1. Generality over different sciences: We seek a function that is applicable to both primarily conceptual models as well as primarily numeric.
2. Ease of computation: The function should not rely too heavily on values that are computationally difficult to obtain. And, once it has its values, it should be rapidly computable.
3. Simplicity of form: There are several competing beliefs for how scientific models should be evaluated. The function's design should be as transparent as possible so that its assumptions are readily comprehended.
4. Smoothness: The function should give similar models similar scores.

We chose these criteria because they are important to our long-term goal of creating a general program to assist a variety of scientists.

Our contributions are the improvements in generality and ease of computation over Thagard's function. Generality is improved in three ways. First, it is adjustable to the tastes of a particular scientist. Second, it is able to handle degrees of inaccuracy. Lastly, it may use statistical arguments as well as proofs. Statistical arguments also improve the ease of computation: the function does not *have* to try to formally prove laws or data using perhaps an undecidable theory. The form of our function, however, is a little more detailed than Thagard's. The smoothness of both of our approaches critically depends upon how the user designs models.

Following Thagard, models have three components: a theory that specifies the details of the model, the data to predict, and a set of laws found from the data and predicted by the theory. The theory and the law set are both composed of assertions in some language. We use first order predicate logic with the data structure extensions of Prolog as our language in this paper. The distinction between which assertions are theory and which are laws is given by Lakatos. He distinguishes between commonly accepted knowledge (the "hard core", *i.e.*, theory) and between more tentatively held knowledge (the "auxiliary hypotheses", *i.e.*, laws) of a given research program

[12][13]. The auxiliary hypotheses are the statements that are not commonly held (*i.e.*, have lower prior probability), and are the main objects that are manipulated during Kuhnian normal scientific discovery [10]. The data is assumed to be in tabular form with associated uncertainties and error bars.

It is simplest to assume that:

1. all measurements are independent of each other,
2. the data influence the choice of law set, and
3. the law set influences the choice of theory assertions.

Figure 1 depicts these assumptions graphically as a Bayesian network.

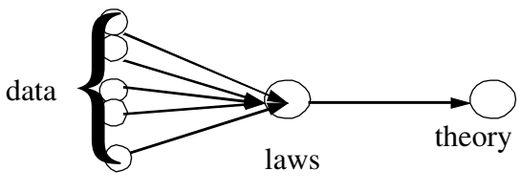


Fig. 1. Bayesian network underlying the relationship between data, laws and theory

We are interested in the most probable total model. We derive the following starting from the Bayesian network of figure 1. Let T denote theory, LS denote a set of laws, and D denote data below:

$$p(T, D) = p(T|D) \cdot p(D) \tag{2}$$

Using Bayes' rule we may re-write this as:

$$= \sum_i p(T) \cdot p(LS_i|T) \cdot p(D|LS_i) \tag{3}$$

The last expression sums over all law sets and is appropriate when there may be disagreement over which law set is best (*e.g.*, several scientists combining their beliefs). However, for an individual scientist, a particular law set may appear much more probable than any of its competitors. In this case we may simplify the expression to:

$$p(T, D) = p(T) \cdot p(LS|T) \cdot p(D|LS) \tag{4}$$

Now we consider the meaning of each term.

The first term of equation 4 tells us the *a priori* probability of a theory, without reference to the law set or data. It encodes the biases on theories. It may be used, for example, to prefer one type of assertion over another. A commonly mentioned bias in science is one for syntactic *simplicity*, which is often measured as the length of an expression in a given language. This first term is the natural place to encode such a bias because this common measure of simplicity is only a function of the length of the expression.

$$p(T) = -\log 2(s(T)) \tag{5}$$

The function s(T) returns a measure of the size of T in some language. The function p(T) uses Shannon information theory to convert from a size to a probability.

We admit that the syntactic length metric is crude. We welcome scientists to redefine p(T) as they choose based upon their own domain knowledge. In defense of this initial estimate of p(T) we note that syntactic metrics: (1) are easy to compute, (2) are well agreed upon as being relevant (if not completely correct), (3) are common to many or all sciences (as opposed to symmetry, for example, which enjoys larger support among physicists than among other scientists), and, (4) would favor syntactically simple theories, which may be easier to comprehend. The last point is especially relevant for *initial* probability distributions, which may return several interesting model space regions that scientists must understand before determining if they warrant further exploration.

The second term tells us how likely the assertions of the law set are given the theory that we have chosen. At one extreme, if all laws are logically entailed by the theory, the term is 1.0 because they must be true (given the theory as premises). It is also 1.0 if the law set is empty because the theory is used to directly compute the data. At the other extreme, the term must be 0.0 if the theory contradicts any statement of the law set. Values in between signify that the law set may or may not follow, depending on specific values of free parameters in the theory. Free parameters are values that the theory refer to that do not have definite values, but distributions over sets of values. Examples include coefficients with standard deviations, and random numbers used during stochastic experiments. In these cases, the second term is set equal to the fraction of the free parameter space in which all of the statements of the law set are found to hold. For random numbers it will be more practical to estimate this value by sampling the space. Laws are limited to refer to the theoretical terms introduced in theories.

The third term measures empirical support and the degree of data coverage by telling us how likely the data are given the statements of the law set and theory. The

same extremes hold when all of the data are logically entailed or some of it is contradicted by the law set or theory. Again, values in between 0.0 and 1.0 represent the fraction of the free parameter space in which the data are observed. Statistical assertions have an implicit free parameter that tells from which data set the statistic was collected. For example, consider two integers, each in the set [0..9], with an average value of 1. The implicit free parameter must denote one of three sets: (1,1), (0,2) or (2,0).

Please consider this (propositional) example. Let our theory be the assertion "a->b", our law be "a" and our data be two occurrences of "b." We would pay the appropriate (perhaps syntactic) price for the theory. The law is not derivable from the theory, so we set its probability to p(a) (the *a priori* probability that free variable A which ranges over "a" and "not(a)" actually is "a"). From our theory and law we may deduce our data with probability 1. If, however, we add assertions "c->a" and "c" to our theory then we have (perhaps) increased theory cost, but the law is now deducible from theory. Thus, the law has probability 1 and has no cost.

A problem with the heuristic function as given is that it has no parameters to be tuned to a particular scientist's preferences. This implies that it always returns the same value for the same arguments. This contradicts our goal of not imposing one ideal form on all scientific models.

Scientists should be able to fine tune the heuristic function, but any adjustment should be general enough to be applicable to all models. Further, we want the number of parameters to be relatively small, both because it will make the function easier to calibrate and because we want to guard against potential abuse by choosing a set of parameters that happen to make one model score well and a similar one score poorly. Our solution was to generalize the function in the following manner:

(6)

$$h_{tm+}\ (T, LS, D) = p(T)^{A} \cdot p(LS|T)^{B} \cdot p(D|LS)^{C}$$

The "tm" signifies that the function is over total models (*i.e*. theory, law set and data) and the "+" reminds us that this a function to maximize (*i.e*., larger values are better). The three parameters A, B and C allow us to independently vary the relative weights of the *a priori* model probability, the law set probability and the data probability.

Instead of maximizing probability, we may view it as minimizing information:

(7)

$$h_{tm-} = A \cdot s(T) - (B \cdot \log 2(p(LS|T))) - (C \cdot \log 2(p(D|LS))$$

The "-" subscript denotes that this function should be minimized.

Equation 7 generalizes original MML equation 1 in two ways. First, equation 1's $-\log p(D|H_i c)$ has been split into two terms, one for both the law set and the data. Both are graded probabilistically. Second, the coefficients A, B and C act as linear weights for the information terms. The linear weights may seem to grossly over generalize equation 1, but it really depends on how they are used. This is discussed in more detail in the next section.

There are two advantages to this weighing approach. First, it conforms to our notions that some sciences value theory conciseness and hard predictions more than others. Set the values of A and C higher in these sciences. Second, it does not allow arbitrary and contrived exceptions to make two similar total models score significantly differently.

Although we have offered a syntactic feature-based approach to specifying a theory's a prior probability, we have not limited scientists to use our function. Further, we admit that this is an iterative approach where probabilities are refined.

Revisiting our criteria we find:

1. Generality is achieved with the adjustable weights, the usage on probabilities of laws instead of counts of "explained facts", the usage of prior distributions instead of "co-hypotheses", and the potential use of proofs or statistical arguments.
2. The ease of computation is limited by our proof or statistical argument method, not by the heuristic.
3. Simplicity is achieved because the form is of a weighted sum with terms for theory, law and data.
4. Smoothness is achieved because lumping all theory together, all laws together and all data together hampers a user's ability to create one model that scores well and another very similar one that scores poorly.

Further generalizations of h_{tm+} and h_{tm-} may be envisioned. Each of the coefficients A, B and C may split into several coefficients A[1..n_1], B[1..n_2] and C[1..n_3]. These finer-grained coefficients may be used to weigh specific aspects of the theory (*e.g.* A[1] for equations, A[2] for decision trees, *etc.*), specific laws of the laws set (*e.g.* B[1] for equations, B[2] for simple logical assertions, *etc.*), and specific types of data (*e.g.* C[1] for spatial measurements, C[2] for temporal measurements, *etc.*)

Using the finer-grained coefficients is justifiable in some cases, like when there are large differences in the precision. For example, in seismology, earthquake times are known with very high precision: to within a few seconds per century. Earthquake locations are known with less precision: to only within tens of kilometers per 40,000 km (the Earth's circumference). Earthquake energies are known with far less precision, frequently only to an order of magnitude. We may want to weigh each type of

data separately, taking into consideration how much precision is given and how much we want this data fit at the expense of other data.

Parameters A, B and C from equations 6 and 7 were not subdivided to simplify analysis and presentation.

5 Experiments and Discussion

This section discusses the rough calibration of the heuristic function to models in two sciences. Geophysics and sociology were chosen because they cover a broad spectrum of acceptable scientific models.

We do *not* evaluate this function by comparing its output with that of IDS, PI, 49er, or Mechem. Which model a scientist believes in given specific data is, at least to some degree, subjective. Rather, we seek a method of calibrating our heuristic such that if it is given examples of models that users like then it can prefer similar models in the future.

The heuristic function's parameters may be calibrated for each science by analyzing its accepted models. Although there are three parameters, we only care about are their relative values. Accordingly, we may set A to 1 and let B and C vary. Equivalently, borrowing from physical chemistry, we can plot B/A versus C/A to create a "phase diagram" that tells which of the various total models are preferred by the heuristic. Each phase diagram constrains the area of each scientific model. This in turn constrains B/A and C/A for all models.

Comparing B/A with C/A makes the linear weights of equation 7 a conservative generalization of equation 1. The plots are primarily a comparison between B and C, and represent a value judgement on how much scientists want their uncertainty in the laws rather than in the data. There is no "correct" answer to this question. As we will see, it varies from scientist to scientist. This also strengthens our argument for an adjustable heuristic function.

If a scientist prefers model X then that scientist should set the parameters to where X is preferred. If the scientist is strongly tempted by model Y, then the scientist should adjust the parameters to be in the region of X but leaning towards that of Y. The scientist may iteratively update the parameter values as new models are evaluated by both the scientist and the heuristic.

Please recall our limited goal: to do an initial search in a large and inhomogeneous space for areas that contain potentially promising models. We do not promise the best models. Also, this may be an iterative process where theory prior probabilities are revised according to previous results.

The Knowledge Base and How It Predicts

The experiments were designed for a variant of the knowledge base discussed in [20]. The knowledge base has two lists of assertions, one for the theory and one for the laws. These assertions describe a standard **is_a** frame hierarchy of knowledge. Assertions may be frame inheritance statements, equations or Prolog-like logic sentences. A Prolog-like resolution engine drives inference, but dedicated code handles frame inheritance and equations for efficiency.

The output of the knowledge base to a given query is either an answer, or FAILURE, signifying no prediction is possible. An information cost accrued by the data when a prediction is wrong or missing. For symbolic values this cost is the Shannon information cost of the prior probability of the recorded answer. Thus, the default model to try to beat is the product of the prior probabilities of each datum. For integers and fixed and floating point values the cost is:[1]

$$-\log 2(DistinctValDiff(predict, record) + 1) \tag{8}$$

where **DistinctValDiff**() returns the number of distinct, representable values between the predicted and recorded values in the attribute's given precision. (For example, if an attribute was limited to multiples of 0.1 then **DistinctValDiff**(0.2,0.4) is 2.) When **predict** is missing then the function is set to its highest value for that attribute.

Sociology Data

This technique requires large amounts of calibration data. We focused on models of family structure because United States Census data on family structure are readily available [4].

Data are not available for specific individuals, but they are summarized in several tables. From these summaries the number of families with 1, 2, 3, 4, 5, and 6 or more "own children" may be calculated for each family type. The family types are married family, male-householder family, female-householder family, married subfamily, male-householder subfamily and female-householder subfamily. Additionally, the number of childless families (but not subfamilies) may be calculated. The term "own children" means children related by birth, marriage or adoption. The U.S. Census

---

[1] Equation 8 corresponds to the last term of equation 7. It defines a maximal probability at the recorded value, and exponential decaying probability above and below that value. This distribution may be replaced by others and is not a critical aspect of this approach.

Bureau switched from “head of house” to “householder” to emphasize the sharing of responsibilities prevalent in modern American families. The term “subfamily” refers to parent(s) who live with other adult(s) who are the householder(s) (*e.g.* their own parent(s).)

We randomly created a database of 10,000 people in proportion to the distribution of household types and number of children computed from the U.S. Census data. This database under represents the number of children a little because the U.S. Census data does not distinguish between 6 or more children. We treated such cases as exactly 6 children. It under represents the number of adults more because we made no attempt to include all cases of adults living with other adults. Our interest is only in predicting where children live as a function of their parents. The database lists each person, their address, and, when the person is a child, their mother and father. Children who did not live with their father got illegal values as their father attribute. This was also done for the mother attribute. All attributes are symbolic.

Sociology Models

After surveying ethnographic reports on 250 societies, Murdock came to the anti-climatic conclusion that the form of families in all societies is of “. . . a married man and woman with their offspring. [17]” (This is a *minimal* family structure because that unit may be embedded in larger structures.)

We take this statement as the theory. We encode it in the structure of the virtual relations of figure 2, augmented with some extra semantics. For example, from the structure of the database we may deduce that all families have one address, one childset, one mother, one father, that a set of children may have 0 or more children, *etc*. The additional rules allow members to inherit selected properties of their families. Predicate **prop(frame,attribute,value)** notes that property **attribute** of **frame** has value **value**.

family	address	childset	mother	father

child	childset	family

$$\forall(child(C) \wedge fam(F) \wedge prop(C, family, F) \wedge prop(F, A, V) \rightarrow prop(C, A, V))$$
$$\forall(fam(F) \wedge prop(F, mother, M) \wedge prop(F, addr, ADDR) \rightarrow prop(M, addr, ADDR))$$
$$etc$$

Fig. 2. Codification of Murdock’s theory

The laws operationalize the theory by making direct predictions about recorded values. For example, assume the child database included address information. We may then note a correlation between a child's address and that of their parent's.

$$\forall(child(C) \wedge mom(M) \wedge fam(F) \wedge prop(C, mom, M) \wedge prop(M, fam, F) \wedge prop(F, addr, A) \rightarrow prop(C, addr, A))$$
$$\forall(child(C) \wedge dad(P) \wedge fam(F) \wedge prop(C, dad, P) \wedge prop(P, fam, F) \wedge prop(F, addr, A) \rightarrow prop(C, addr, A))$$

Fig. 3. Codification of potential Murdock laws (atoms *mother*, *father* and *family* have been abbreviated as *mom*, *dad* and *fam*)

The competing sociological model is due to Adams [1]. After examining Latin American and some ethnic societies, Adams concluded that the evidence for the nuclear families as described by Murdock was "marginal at best" [14]. Instead he proposed the mother-child dyad as the primary unit. This new model is created by removing the **father** attribute, or merely disallowing its use in proofs. We also delete the **father** law mentioned in figure 3 from the law set.

We bound the parameters by considering two unacceptable models at opposite extremes. The first is the "data" model. It uses neither theory nor laws to predict values. It merely reflects the prior probability of any one value. The second is the "theory" model. It explicitly memorizes each value individually as a statement in the theory. It has neither general statements nor laws, and overfits the data.

Table 1 gives the sizes of the each component of each total model. Both Murdock's and Adams' models must memorize adult addresses. Adams' must also memorize those of children who live with their fathers but not mothers. The law sentences in figure 3 logically follow from theory so they have size 0. Unfortunately, the zero size forbids the constraining of the B parameter by this experiment.

Table 1. Sizes of sociological models

Model	**Abbr**	**Theory**	**Law**	**Data**
data	d	0	0	107637
Adams	a	240	0	79582
Murdock	m	480	0	77739
theory	t	960960	0	0
Adams'	A	240	23429	77739

Figure 4a gives the "phase diagram" plot of data. Where a model out scores all others its abbreviating letter appears in the parameter space. log2(C/A) is plotted on the X axis and log2(B/A) on the Y.

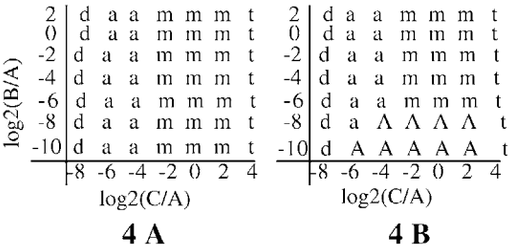


Fig. 4. Sociology model "phase diagrams"

To place bounds on B we consider adding the **father** sentence to Adams' law set. However, we cannot prove it from our theory. Therefore, we accept **father** in the model as a free variable with its (data-specified) prior probabilities. This results in a model with the equivalent predictive power of Murdock's. It can now predict the addresses of children living with only their fathers. The price we pay is Shannon information cost of the prior probability of each usage of the **prop(Child,father,Father)** predicate for these predictions. See Adams' in table 1. The revised "phase diagram" with Adams' new model is plot in figure 4b.

Geophysics Data

We obtained data from the United States Geological Survey's National Earthquake Information Center. We retrieved all recorded earthquakes in the catalog in a rectangular box from 139E to 162E and from 41N to 55N from 1976 to 2000. The Kuril subduction zone, the Japanese island of Hokkaido, and the Kuril island chain are the most prominent geophysical features in this area. Non-tectonic events were removed and the remaining ones were fit to a great circle. This great circle was taken to be the "length" of the fault and events greater than 512 km from it were removed. The time, distance-along-fault, (signed) distance-from-fault and depth of the remaining 11031 events were entered into our earthquake database.

Geophysics Model

In the theory of plate tectonics, a *subduction zone* is a region where one (oceanic) plate sinks beneath another (continental) plate. A *Wadati-Benioff zone* is the seismically active portion of this interface [2][23].

A Wadati-Benioff zone may be modeled as a plane that increases in depth the further one goes into the continental plate. We did so by stating the assertions of figure 5 in the theory where the slope and intercept were found by least-squares fit.

$$DistFromfault = slope \times depth + intercept$$

inherit(kuril_quakes,slope,1.05682).
inherit(kuril_quakes,inter cept, -85.9936 km).

Fig. 5. The theory of the planar Wadati-Benioff zone model**.**

The law set was left empty. As before, the "data" model did not try to predict, and the "theory" model overfit by memorization. The results are given in Table 2 and are plotted in Figure 6a.

Table 2. Sizes of geophysical models**.**

Model	**Abbr**	**Theory**	**Law**	**Data**
data	d	0	0	97750
planar	p	618	0	63904
theory	t	1369230	0	9775
aftershock	a	618	13759	63103

The non-zero entry for the theory model's for data size is due to round off error. That is, there is a slight difference between the decimal recording of the values logical assertions that comprise the theory (which have a fixed number of significant digits given by the precision of the values), and the binary recording of the values in the database.

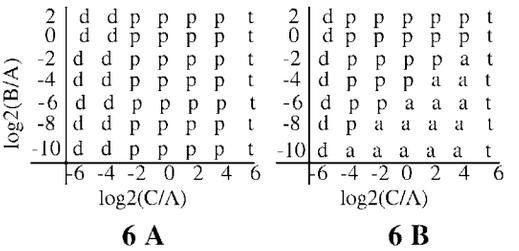


Fig. 6. Geophysics model "phase diagrams"

To place bounds on B we add a law to the planar model. When a particular aftershock labelling procedure is used there is an average of a 43.5 km distance between

an aftershock and its mainshock. Encoding this as a law permits better predictions of some distances. We include no theory to predict aftershocks, only an empirical procedure for labeling them after the fact. Therefore, we let **mainshock** be a free variable. The aftershock model results are given in Table 2 and in Figure 6B.

We now evaluate our heuristic with the criteria in section 4. Recall, they were (1) generality, (2) ease of computation, (3) simplicity of form, and (4) smoothness. The function is general because it was applied to symbolic sociology and numeric geophysics with equal ease, and because it has been applied to a domain where predictions have varying degrees of accuracy. Its ease of computation is limited by the ability to predict data, prove (or argue for) laws, and know data distributions. Also, its weighted sum form is simple.

The function's "smoothness," its ability to give similar models similar scores, is limited by how honest people are with the law set. When some condition is true over the whole parameter space one could move it from theory to laws to avoid paying the syntactic cost. This is against the philosophy of this approach. Also, trying to estimate data distributions when there is little data may be a serious problem. Distributions may be used as "fudge factors" to vary a model's score on the B/A axis. However, a potential advantage is that it will force such assumptions to be explicitly stated.

We do not argue for one particular ratio for C/A or B/A. Rather, we seek a method for calibration. That said, we note that both geophysics and sociological had similar C/A bounds. Having B be too great may lead to "overfitting" the laws to the theory and ruling out yet unknown secondary effects. For discovery it may be best to fix A and C and let B vary as the model becomes more refined. This is another study.

Note that this was truly a test of scientific *rediscovery*. Both the sociology and the geophysics theories were applied to new data. Neither Adams nor Murdock were trying to fit U.S. demographics for 1998. Benioff stated his hypothesis after examining events from S. America and Hindu-Kish, not the Kurils. (Wadati probably had data for Honshu, not the Kurils.)

6 Conclusion

Scientists have different opinions on what the same data entails. To ignore that is to ignore the history of science. We have developed a heuristic function that takes some of these differences into account, and may be calibrated to a particular scientist, along our given axes. This heuristic function is a generalization of single model family parameter finding MML. It generalizes MML in a principled fashion to consider how much faith to put in laws versus data. Our approach also extends [23] to be applied to scientific discovery. It is general and has been applied to both symbolic and numeric scientific models.

We do not claim to have solved the whole scientific model preferencing problem. Serious limitations remain including (1) the specification of the original model prior probability, (2) the inhomogeneity of the search space, and (3) the fact that the "most probable" model is not necessarily the best one. The purpose of this heuristic is to help scientists identify interesting regions in the model space, *i.e*., models that are the immediate neighbors of their favorite models in the B/A-C/A plots. This is an initial step of an iterative process.

Computer scientists might believe that a heuristic function could not sufficiently constrain search in a domain as rich as scientific discovery. However, the heuristic function is only part of the search algorithm. The search algorithm may employ rules to suggest when to apply scientific operators (*e.g*., [11]), or may use metalearning to discover which operators are best in a particular domain. Preliminary results from rediscovery in geophysics show that rules and metalearning may be combined or employed separately to significantly speed scientific discovery [20].

Acknowledgments

I thank my geophysicist Larry Ruff for his patience, my former advisors John Laird and Nandit Soparkar, and the National Physical Science Consortium and the Rackham Merit Fellowship for funding.

References

1. Adams, R.N. 1960. An inquiry into the nature of the family. p 30-49 in Dole, G. and Carneiro, R.L. (eds.), *Essays in the Science of Culture: In Honor of Leslie A. White*. Thomas Y. Crowell. New York.
2. Benioff, H., 1948. Earthquakes and rock creep. *Geol. Soc. Am. Bull.*, 59, p. 1391.
3. Buchanan, B., Phillips, J. 2001. Towards a computational model of hypothesis formation and model building in science. *Model Based Reasoning: Scientific Discovery, Technological Innovation, Values*. Kluwer.
4. Casper, L., Bryson, K. 1998. *Current Population Reports: Population Characteristics: Household and Family Characteristics. March 1998 (Update)*. United States Census Bureau.
5. Cheeseman, P. 1995. On Bayesian model selection. In Wolpert, D. (ed.) *The Mathematics of Generalization: Proceedings of teh SFI/CNLS Workshop on Formal Approaches to Supervised Learning*. Addison-Wesley: Reading, MA.
6. Forbus, K., 1985, Qualitative process theory, in *Qualitative reasoning about physical system*s, D. Bobrow, ed., MIT Press: Cambridge, Mass.
7. Fuller, S. 1993. *Philosophy of Science and its Discontents, Second Edition*. Guilford Press, New York.

8. Georgeff, M.P. and Wallace, C.S. 1984. A general selection criterion for induction inference. In *Proceedings of the European Conference on Artificial Intelligence*, p. 473-482. Elsevier: Amsterdam.
9. Korf, R.E. 1988. Search: A Survey of recent results. In H.E. Shrobe (Ed.), *Exploring Artificial Intelligence: Survey Talks from the National Conferences on Artificial Intelligence* (pp. 197-237). Morgan Kaufman.
10. Kuhn, T. 1962. *The Structure of Scientific Revolutions*. University of Chicago: Chicago.
11. Kulkarni, D. and Simon, H. 1988. The processes of scientific discovery: the strategy of experimentation, *Cognitive Science*, vol. 12, p. 139-175.
12. Lakatos, I. 1970. Falsification and the methodology of scientific research programmes. In Lakatos, I. and Musgrave, A. (ed.) *Criticism and the growth of knowledge*. Cambridge University Press: Cambridge.
13. Lakatos, I. 1971. History of science and its rational reconstructions. In Buck, R.C. and Cohen, R.S. (ed.) *Boston Studies in the Philosophy of Science*. vol 8, p 91-135. Reidel: Dordrecht.
14. Lee, G. 1977. *Family Structure and Interaction: A Comparative Analysis*. J.B. Lippincott. Philadelphia.
15. McAllister, J. 1996. *Beauty and Revolution in Science*. Cornell University: Ithaca.
16. Michalewicz, Z., Fogel, D. 2000. *How to Solve It: Modern Heuristics*. Springer-Verlag. Berlin.
17. Murdock, G.P. 1949. *Social Structure*. The Free Press. New York.
18. Nordhausen, B., Langley, P., 1987, Towards an integrated discovery system, in *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, Milan, Italy.
19. Nordhausen, B., Langley, P., 1990, An integrated approach to empirical discovery, in Shrager J, and Langley, P. (ed.) *Computational Models of Scientific Discovery and Theory Formation*. Morgan Kaufmann, San Mateo.
20. Phillips, J. 2000. *Representation Reducing Heuristics for Semi-Automated Scientific Discovery*. Ph D. Thesis, University of Michigan.
21. Rissanen, J. 1978. Modeling by shortest data description. *Automatica*, 14, p. 45-471.
22. Sleep, N., Fujita, K. 1997. *Principles of Geophysics*. Blackwell Science. Malden.
23. Thagard, P. 1988. *Computational Philosophy of Science*, MIT Press, Cambridge MA.
24. Wallace, C.S., and Freeman, P.R. 1987. Estimation and inference by compact encoding. *J. Roy. Stat. Soc., Series B*, 49, p 233-265.

25. Valdes-Perez, R. 1995. Machine discovery in chemistry: new results. *Artificial Intelligence*, 74(1), p 191-201.
26. Zembowicz, R. and Zytkow, J. 1996. From contingency tables to various forms of knowledge in databases, in: *Advances in Knowledge Discovery and Data Mining*, Fayyad et al (eds.) AAAI Press, San Mateo.
27. Zytkow, J. and Zembowicz, R. 1993. Database exploration in the search for regularities, *J. Intelligent Information Systems*, 2:39-81.

Efficient Local Search in Conceptual Clustering

Céline Robardet and Fabien Feschet

Laboratoire d'Analyse des Systèmes de Santé
Université Lyon 1
UMR 5823, bât 101, 43 bd du 11 nov. 1918
69622 Villeurbanne cedex
FRANCE
robardet@univ-lyon1.fr

Abstract. In this paper, we consider unsupervised clustering as a combinatorial optimization problem. We focus on the use of *Local Search* procedures to optimize an association coefficient whose aim is to construct a couple of conceptual partitions, one on the set of objects and the other one on the set of attribute-value pairs. We present a study of the variation of the function in order to decrease the complexity of local search and to propose stochastic local search. Performances of the given algorithms are tested on synthetic data sets and the real data set *Vote* taken from the UCI Irvine repository.

Keywords: Unsupervised conceptual clustering, optimization procedure, local search.

1 Introduction

In the early steps of knowledge discovery from large databases, structuring data appears as a fundamental procedure which permits to better understand the data and to define groups with regards to an a priori similarity measure. This is usually referred to clustering in the unsupervised learning context. The data are composed of a set of objects described by a set of attributes such that each object owns a value on every attributes. In classification/regression, we have a target attribute which can be used to construct the groups. Knowledge discovery can be done through the learning of rules which explain the values on the target attribute using the other attributes. In this way, to each group of objects is associated a set of attribute-value pairs [Rak97]. When no prior information is available, clustering procedures can be used to discover the underlying structure of the data. They construct a partition on the set of objects such that *most* similar objects belong to a same cluster whereas *most* dissimilar ones belong to different groups. Hence, those procedures synthesize the data into few clusters.

One of the key points in clustering is the a priori definition of similarity. When dealing with numerical attributes, it is usual to relate the similarity between two objects with their distance. Clustering is then reduced to the determination of groups minimizing the intra-cluster similarity and maximizing the inter-clusters one. For instance, in the K-MEANS algorithm [JD88,CDG+88], Euclidean distances between representative vectors of objects are used. This can

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 323–335, 2001.

also be extended to ordinal data and even to symbolic one but distances become less representative in this case. Instead, probabilistic representations are preferred. The difference between the probabilities of appearance of an attribute-value pair on the whole set of objects and its restriction on the set of objects belonging to a particular cluster is used to guide the search for a *good* partition. It is a trade-off between intra-class similarity and inter-class dissimilarity of the objects. For example, in the COBWEB algorithm [Fis87,Fis96], the category utility function is used as an objective function. It is a weighted averaging of the well known GINI index without fixing the number of clusters. Other methods like AUTOCLASS [CS96] also use bayesian classification, modeling objects by finite mixture distributions.

Another key point in clustering is the optimization procedure. The cardinality of the set of all possible partitions increases exponentially with the size n of the set of objects, which leads to use fast but often rough heuristics. In the K-MEANS algorithm, a heuristic based on the principle of reallocation, is used. At each step, cluster centroids are computed and each object is assigned to the cluster whose centroid is the closest. After few such steps the procedure stops to improve the partition. But unfortunately, the algorithm makes only local changes to the initial partition and thus typically gets trapped in the first local minimum. COBWEB method uses an incremental procedure which classifies objects one by one. For each object, the procedure evaluates the two following options: classifying the object in one of the existing clusters or creating a new one containing only this object. The operation which leads to the most important increase in the function is considered. The main drawback of this heuristic is that it often constructs a local optimum which is dependant on the order of the objects in the incremental process. In AUTOCLASS, optimization is done for maximum posterior parameters (MAP) with the EM algorithm. In fact, among a set of models, constituted of *a priori* number of clusters and probability distributions functions, the method consists in estimating some parameters using the EM algorithm and choosing the best model using a MAP estimator.

Optimization can be global or local. The first one is usually unreachable and the second is very sensitive to initial conditions. Popular methods like *Tabu Search* or *Genetic Algorithm* are widely used without knowing clearly how they work. In this paper, we restrict to local optimization procedure and more precisely on the simplest one that is the *local search* procedure. Local optimization seems to be a promising method for clustering since it has provide good results at a low cost in lots of combinatorial optimization problems. We base our study on a variational approach of an objective function which is described in section 2. Variations of the function through elementary modifications are studied in section 3 where a single model of modification is given. This permits us to introduce five stochastic optimization procedures which are experimentally studied on two different data sets. The first one is an artificial data set and the second one is the Vote data from the UCI Irvine repository. We then propose some conclusions and future works.

2 Clustering Method

To strengthen the semantic knowledge held by partitions, we study an algorithm for the construction of two linked partitions, one on the set of objects and the other one on the set of attribute-value pairs; we call this couple a *bi-partition*. Similar methods have already been proposed. We can cite the methods of data reorganization [MSW72,SCH75] which consist in permuting rows and columns of a data table on the base of a distance to minimize. Another one is the *simultaneous clustering algorithm* [Gov84]. It consists in searching a couple of partitions in *a priori* K and L clusters and an *ideal* binary table of dimensions $K \times L$ such that the gap between the initial data table structured by the two partitions and the *ideal* table is minimized. Those two procedures have important drawbacks. The first methods do not produce partitions which must be constructed by the user. The second one determines a couple of partitions with *a priori* fixed numbers of clusters. Furthermore, the resulting couple of partitions is often far from the global optimum.

To enforce the knowledge contribution brought by the bi-partition, we favor couples of partitions which follow the following property,

Property: The functional link, which restores one partition on the basis of the knowledge of the second one, must be as strong as possible. Furthermore, both partitions must have the same numbers of clusters.

To evaluate the quality of a bi-partition regarding this property, we construct a function over $\mathcal{P}_\mathcal{O} \times \mathcal{P}_\mathcal{Q}$, where $\mathcal{P}_\mathcal{O}$ is the set of partitions on the set of objects, and $\mathcal{P}_\mathcal{Q}$ is the set of partitions on the set of attribute-value pairs. This function must follow some properties [Rak97,RF01] to be adapted to the clustering structure, such as the independence upon clusters permutations or the ability to treat bi-partitions having partitions with different numbers of clusters, etc. Theses properties are partially checked by association measures, which have been built to evaluate the link between two qualitative attributes X and Y, which are considered as partitions upon a same set. The association measures are widely used in supervised clustering [LdC96], whereas few unsupervised clustering algorithms used them [MH91]. We propose [RF01] to use an adaptation of the τ_b measure construct by Goodman and Kruskal [GK54], which we call $\tau_\mathcal{Q}$,

$$\tau_\mathcal{Q} = \frac{\sum_i \sum_j \frac{p_{ij}^2}{p_{i.}} - \sum_j p_{.j}^2}{1 - \sum_j p_{.j}^2}$$

We name $\tau_\mathcal{O}$ the above measure obtained when exchanging the attributes[1]. We denote by $p_{i.}$ (resp. $p_{.j}$) the frequency estimator of the probability associated to the attribute-value pair i (resp. j) of the X (resp. Y) attribute, and by p_{ij} the frequency estimator of the probability that an attribute-value pair i of the attribute X, and the attribute-value pair j of Y arisen simultaneously. The $\tau_\mathcal{Q}$

[1] $\tau_\mathcal{O}$ is used to determine an adequate partition on $\mathcal{P}_\mathcal{O}$ and $\tau_\mathcal{Q}$ is used to obtain an adequate one on $\mathcal{P}_\mathcal{Q}$

coefficient evaluates the *proportional reduction in error* given by the knowledge of the attribute X on the prediction of Y. It takes into account all the structure of the distribution when estimating the variation on the prediction. Using this measure, we do not need to fix the number of clusters in the partitions. It measures how the knowledge of the partition P of $\mathcal{P}_\mathcal{O}$ improve the prediction of the cluster of an attribute-value pair in a partition Q of $\mathcal{P}_\mathcal{Q}$, knowing the cluster(s) of P which possess objects described by the attribute-value pair. The measure is normalized and consequently none of the discrete or the single-cluster partitions are favored. Moreover, some experiments have been realized by M. Olszak [Ols95] and also by us [RF01]. They consist in comparing several association measures with regard to different synthetic data sets. In both studies, the authors find that the $\tau_\mathcal{Q}$ has an appropriate behavior.

To overcome the fact that our two partitions are not based on a same set, we build a co-occurrence table. In the data, each object is described by h attributes V_i such that $V_i : \mathcal{O} \to \mathrm{dom}_i$. $\mathcal{Q} = \biguplus_{i=1}^{h} \mathrm{dom}_i$ is the set of all attribute-value pairs, differentiating each attribute value of the different attributes. The co-occurrence table between a partition $P = (P_1, \ldots, P_K)$ on the set $\mathcal{O}$ of objects and a partition $Q = (Q_1, \ldots, Q_K)$ on the set $\mathcal{Q}$, is $(n_{ij})_{i,j}$ with

$$n_{ij} = \sum_{x \in P_i} \sum_{y \in Q_j} \sum_{i=1}^{h} \delta_{V_i(x),y}$$

where δ is the Kronecker[2] symbol. Consequently, we replace the previous p_{ij} (resp. $p_{i.}$) notation by $\frac{n_{ij}}{n_{..}}$ (resp. $\frac{n_{i.}}{n_{..}}$) where $n_{i.} = \sum_j n_{ij}$ and $n_{..} = \sum_i \sum_j n_{ij}$

To determine the best bi-partition, we search a bi-partition which maximizes the $\tau_\mathcal{Q}$ and $\tau_\mathcal{O}$ measures. The problem is now to find an adequate optimization procedure, remembering that we are confronted to a combinatorial optimization problem. Note that the search space $\mathcal{P}_\mathcal{O} \times \mathcal{P}_\mathcal{Q}$ is huge (exponential in n)

$$\sharp\left(\mathcal{P}_\mathcal{X}\right) = \sum_{c=1}^{m} \frac{1}{c!} \sum_{i=1}^{c} (-1)^{c-i} \binom{c}{i} i^n \quad \textit{with} \quad \sharp\left(\mathcal{X}\right) = m \quad \textit{and} \quad \mathcal{X} = \{\mathcal{O}, \mathcal{Q}\}$$

Consequently exhaustive or potentially exhaustive search procedures, like the *Branch and Bound*, is unrealistic in terms of time efficiency. Using others procedures, we have no guarantees the obtained solution is a global optimum. Choosing a local optimization method is a trade-off between computation cost and quality of the result.

3 Local Search

We consider general purpose methods which are based on the definition of the neighborhood of a given partition. At each step, a new solution is chosen among the neighborhood of the previous one, such that the algorithm converges towards

[2] $\delta_{V_i(x),y} = 1 \quad \text{if} \quad V_i\left(x\right) = y, \quad \delta_{V_i(x),y} = 0 \quad \textit{otherwise}$

at least a local optimum. Generating several possible solutions at each step allows to direct the search to the candidates which most improve the function. The main difficult point is to determine how to construct an efficient neighborhood sufficiently rich and with a tractable complexity. Recent works [FK00,GKLN00] attempt to apply *Local Search* algorithm to clustering problem. [GKLN00] propose six operators for generating a partition starting from another. They apply those operators first successively, and then stochastically following their frequency of improving the function. They observe that the second algorithm is more robust than the first one. [FK00] couples together *Local Search* and K-MEANS algorithms. The neighborhood function consists in randomly swapping a cluster centroid by another object and then applying the K-MEANS procedure. This procedure is less dependant on the initialization of the algorithm and provide robust results. Both papers introduce randomness in the generating neighborhood process and observe increase in the quality of the results.

Local Search is often compared with *Tabu Search*, *Genetic Algorithms*, and *Simulated Annealing* which attempt to obtain a possibly global optimum without visiting all possible solutions. *Tabu Search* consists in choosing a better solution than the current one when it exists, and to accept sub-optimal solution otherwise. A Tabu list prevents to return to a candidate recently evaluated. The procedure can thus pass through local optimum but often with a high computing time. *Simulated Annealing* relies on a stochastic process which allows to escape from local optima. Solutions which improve the objective function are not necessarily kept. The selection process consists in taking solutions regarding their associated probability. This probability increases for solutions improving the function. But the probability is also influence by a global parameter called *temperature* which gradually decreases to force the convergence of the algorithm to an optimum. Whereas other methods generate a unique new solution at each step, the particularity of *Genetic Algorithms* [Rud94] is to generate a set of best solutions, called population, at each step. The neighborhood of the population is defined using genetic operators such as *reproduction*, *mutation* and *crossover*. New candidates which surpass their parents are always maintained, which guarantees the convergence to a good solution. [BRE91,Col98] apply such algorithms to clustering problems.

4 Variational Approach

For using local optimization procedures we usually define operators. Then we apply them on the current solution to generate the neighborhood. After that, we compute the measure on each member of the neighborhood and compare the value with the one obtained on the current solution. This procedure is expensive in memory space used and computing time. In our problem, computing the measure on a new partition might require to duplicate the co-occurrence table and consequently to double the memory space used, which is a drawback for the scalability of the method. Furthermore, the complexity for evaluating the $\tau_{\mathcal{Q}}$

measure is in $\mathcal{O}(p \times q)$, with p denoting the number of clusters of P and q the ones of Q. This cost is multiplied by the cardinal of the neighborhood.

To overcome those drawbacks, we propose a variational approach for evaluating the objective function. We define three operators for generating neighboring partitions. Those operators are the *transfer* of one element from a cluster to another, the *split* of a cluster into two and the *merging* of two clusters into one. Those operators constitute a complete generating system because what ever the current partition is, we can reach each of the other ones by applying a finite number of such operators. We evaluate the variation on the $\tau_{\mathcal{Q}}$ measure when modifying the current partition by one of the three operators.

We first consider the variation on $\tau_{\mathcal{Q}}$ when transferring, on the partition Q, one attribute-value pair y from a group denoted by b to another denoted by e. Given than each cluster of Q is linked to a column of the co-occurrence table, the transfer of y from Q_b to Q_e generates the moving of a quantity λ_i^y from the cell on row i and column b to the one on row i and column e. Let us denote by n_{ij} the elements of the old co-occurrence table, and by m_{ij} those of the new one. The transfert of y induces the following equations between n_{ij} and m_{ij}

$$\begin{array}{lll} m_{ib} = n_{ib} - \lambda_i^y & ; & m_{ie} = n_{ie} + \lambda_i^y \\ m_{ij} = n_{ij} & & \text{otherwise} \end{array} \tag{1}$$

The variation of $\tau_{\mathcal{Q}}$ given by the transfert is then

$$\tau_{\mathcal{Q}}^{old} - \tau_{\mathcal{Q}}^{new} = \frac{\sum_i \sum_j \frac{n_{ij}^2}{n_{i.}n_{..}} - \sum_j \frac{n_{.j}^2}{n_{..}^2}}{1 - \sum_j \frac{n_{.j}^2}{n_{..}^2}} - \frac{\sum_i \sum_j \frac{m_{ij}^2}{n_{i.}n_{..}} - \sum_j \frac{m_{.j}^2}{n_{..}^2}}{1 - \sum_j \frac{m_{.j}^2}{n_{..}^2}}$$

Simplifying using equations (1), we obtain

$$\tau_{\mathcal{Q}}^{old} - \tau_{\mathcal{Q}}^{new} = \frac{I \times \left(\sum_i \frac{2\lambda_i^y}{n_{..}n_{i.}} \left[n_{ib} - n_{ie} - \lambda_i^y \right] \right) + C \times \left(\frac{2\lambda^y}{n_{..}^2} \left[n_{.e} - n_{.b} + \lambda^y \right] \right)}{I^2 - \frac{2}{n_{..}^2} \lambda^y I \left(n_{.e} - n_{.b} + \lambda^y \right)}$$

where $\lambda^y = \sum_i \lambda_i^y$, and I and C are the following constants with respect to b and e,

$$I = 1 - \sum_j \frac{n_{.j}^2}{n_{..}^2} \qquad C = 1 - \sum_i \sum_j \frac{n_{ij}^2}{n_{i.}n_{..}}$$

The transfer of several attribute-value pairs in a same movement leads to the same expression. Indeed, considering the transfer of a set $\mathcal{S}$ of attribute-value pairs, we compute $\left(\lambda_i^{\mathcal{S}}\right)$ vectors as follows

$$\lambda_i^y = \sum_{x \in P_i} \sum_{i=1}^{h} \delta_{V_i(x),y} \quad \textit{and} \quad \lambda_i^{\mathcal{S}} = \sum_{y \in \mathcal{S}} \lambda_i^y$$

Consequently, $\left(\lambda_i^{\mathcal{S}}\right)$ vectors are linear combinations of the (λ_i^y), and transferring a single attribute-value pair or a set of them is evaluated by the same expression.

Furthermore, the fusion of two clusters into a single one can be considered as a transfer of all attribute-value pairs of a cluster into another one, and thus leads to empty the first cluster. The computational expression is similar of the transfer's one. When two columns b and e are merged into the e one, we have the following expression,

$$\tau_{\mathcal{Q}}^{old} - \tau_{\mathcal{Q}}^{new} = \frac{I \times \left(\frac{-2}{n_{..}} \sum_i \frac{\lambda_i^b}{n_{i.}} n_{ie} \right) + C \times \left(\frac{2\lambda^b}{n_{..}^2} n_{.e} \right)}{I^2 - I \frac{2}{n_{..}^2} \lambda^b n_{.e}}$$

Splitting a cluster into two is also a transfer like operation. It can be view as a transfer of a set $\mathcal{S}$ of attribute-value pairs into a new cluster. When a column b is split into an e and b ones, the variation of $\tau_{\mathcal{Q}}$ is:

$$\tau_{\mathcal{Q}}^{old} - \tau_{\mathcal{Q}}^{new} = \frac{I \times \left(\frac{2}{n_{..}} \sum_i \frac{\lambda_i^{\mathcal{S}}}{n_{i.}} \left[n_{ib} - \lambda_i^{\mathcal{S}} \right] \right) + C \times \left(\frac{2\lambda^{\mathcal{S}}}{n_{..}^2} \left[\lambda^{\mathcal{S}} - n_{.b} \right] \right)}{I^2 - I \frac{2}{n_{..}^2} \lambda^{\mathcal{S}} \left(\lambda^{\mathcal{S}} - n_{.b} \right)}$$

Similar expressions are found when moving a subset of objects for one row to another on the $\tau_{\mathcal{O}}$ measure.

Through the above expressions we show that the variation on the $\tau_{\mathcal{Q}}$ measure can be evaluated using the co-occurrence table, for the evaluation of the n_{ij} parameters, and the data table, for the computing of the λ_i expressions. The partition itself is not taken into account for computing the variations. Furthermore, we have shown that the three different operators lead to a unique expression of the $\tau_{\mathcal{Q}}$ variation, which we denote by $\Delta\left(\left(\lambda_i^{\mathcal{S}}\right), b, e\right)$. The fusion and merge are particular cases of the transfer modification. Computing $\Delta\left(\left(\lambda_i^{\mathcal{S}}\right), b, e\right)$ has a lower computational complexity than evaluating $\tau_{\mathcal{Q}}^{old} - \tau_{\mathcal{Q}}^{new}$. In the variational approach, the evaluation of the first partition is in $\mathcal{O}(p \times q)$ because we need to compute the constant C. Then, when the constants I and C are fixed, the complexity for evaluating a new partition is in $\mathcal{O}(\max(p, q))$. When we need to upgrade the constants I and C, it takes $\mathcal{O}(1)$ and $\mathcal{O}(p)$ respectively. Consequently, we reduce the complexity from $\mathcal{O}(p \times q)$ to $\mathcal{O}(\max(p, q))$, except for the first evaluation.

Globally, the dimension of the problem is reduced. It is now expressed as a function of the elementary vectors (λ_i^y), with $\lambda_i^y = \sum_{x \in P_i} \sum_{i=1}^{h} \delta_{V_i(x),y}$ for all attribute-value pairs y. All $\left(\lambda_i^{\mathcal{S}}\right)$ vectors can be generate from the elementary vectors (λ_i^y) as follows

$$\left(\lambda_i^{\mathcal{S}}\right) = \sum_{y \in \mathcal{Q}} \epsilon_y \left(\lambda_i^y\right) \quad \epsilon_y \in \{0, 1\}$$

The problem is now to find a way to determine the $\left(\lambda_i^{\mathcal{S}}\right)$ vectors which lead to the most important increase in the measure. In the next section, we propose five algorithms which differ from their way to choose such vectors.

5 Algorithms

Using the previous variational approach into a *Local Search* procedure leads to the following deterministic algorithm

```
For each cluster P_b do
    For each attribute-value pair y of P_b do
        For each cluster P_e ≠ P_b do
            Compute Δ((λ_i^y), b, e)
        End For
        If (min Δ < 0) then
        Modify the co-occurrence table
    End For
End For
```

At each step, we consider one attribute-value pair per cluster and try to transfer it to another cluster. We modify the co-occurrence table for the transfer with highest negative decrease.

It is well known that randomness usually increases the performance of deterministic algorithm. Stochastic optimization can be considered as a random walk above the set of all partitions. If this search is guided to be attracted by high values of some measure on the partitions, the probability to visit the partitions with global maximum value are increased [FK00]. We thus propose four randomized versions depending on which *For* loop is randomized in the deterministic version.

Stochastic 1 algorithm	***Stochastic 2 algorithm***

Randomly choose a cluster P_b
Randomly choose y,
an attribute-value pair
For each cluster $P_e \neq P_b$ do
Compute $\Delta\left(\left(\lambda_i^y\right), b, e\right)$
End For
If $(\min \Delta < 0)$ then
Modify the co-occurrence table
	Randomly choose a cluster P_b
	Randomly choose a subset $\mathcal{S}$ in P_b
	For each cluster $P_e \neq P_b$ do
	Compute $\Delta\left(\left(\lambda_i^{\mathcal{S}}\right), b, e\right)$
	End For
	If $(\min \Delta < 0)$ then
	Modify the co-occurrence table

Stochastic 3 algorithm	***Stochatic 4 algorithm***

For each cluster P_b do
Randomly choose y in P_b
For each cluster $P_e \neq P_b$ do
Compute $\Delta\left(\left(\lambda_i^y\right), b, e\right)$
End For
If $(\min \Delta < 0)$ then
Modify the co-occurrence table
End For
	For each cluster P_b do
	Randomly choose a subset $\mathcal{S}$ in P_b
	For each cluster $P_e \neq P_b$ do
	Compute $\Delta\left(\left(\lambda_i^{\mathcal{S}}\right), b, e\right)$
	End For
	If $(\min \Delta < 0)$ then
	Modify the co-occurrence table
	End For

Those algorithm are several combinations between randomness and deterministic choice of the cluster and the attribute-value pair(s) to modify. Note that even and odd versions differ by the choice of one or a subset of attribute-value pairs. In the two first algorithms, the cluster, from which y or $\mathcal{S}$ is removed, is chosen randomly, whereas for the two last ones all the clusters are examined. In all those algorithms, the best ended cluster is chosen after examining all the possible ones.

6 Experimentation

To optimize a bi-partition, we successively execute the algorithm with $\tau_{\mathcal{Q}}$ as objective function which leads to improve the partition Q of $\mathcal{P}_{\mathcal{Q}}$ and then we apply the same algorithm with $\tau_{\mathcal{O}}$ objective function and thus improve the partition P of $\mathcal{P}_{\mathcal{O}}$. We must underline the fact that modifying Q (resp. P) greatly influences $\tau_{\mathcal{Q}}$ (resp. $\tau_{\mathcal{O}}$) and in a less extent influences also $\tau_{\mathcal{O}}$ (resp. $\tau_{\mathcal{Q}}$). This explains the fact that on some of the following graphics we observe a decrease in the measure.

We first apply those algorithms on a perfect synthetic data set which contains 200 objects and 30 attributes with 5 different values each. This data set is composed of 5 blocks of homogenous data, composing a bi-partition into 5 clusters. Starting from the discrete partition (Fig.1 left) or from a random partition (Fig.1 droite), we apply the five algorithms on this data set. On Fig.1, the value of $\tau_{\mathcal{Q}}$ is plotted at each step.

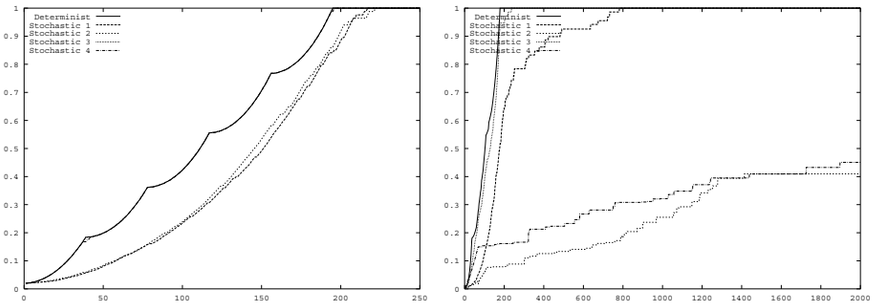


Fig. 1. Perfect synthetic data set, starting from the discrete partition (left), or from a random one (right)

On the synthetic data set, we observe that the deterministic and the third stochastic procedures find in fewer steps the optimal partition than the other procedures. This can be explained by the fact that in those procedures all possible clusters P_b and all possible cluster P_e are evaluated and that at each step the best movement for a given single y is chosen. The first stochastic procedure is also really impressive. When the first partition is the discrete one (see Fig.1 left), it

has the same behavior than the deterministic procedure. When the first partition is constructed randomly (see Fig.1 right), it takes more steps to find the goal partition. The second and the fourth stochastic procedures are the slowest. They rely on a randomly choice of a subset of attribute-value pairs. When the subset is composed of dissimilar attribute-value pairs, the procedure can not improve the value of the measure. This explains the fact that those procedures have better performances on the left graphic, when the first partition is the discrete one and consequently the possible subsets $\mathcal{S}$ are of small cardinality.

To simulate a more realistic case, we randomly introduce some noise in the data set (see Fig.2 which shows the $\tau_{\mathcal{Q}}$ value for each iteration with 10% (left) and 30% (right) of random noise).

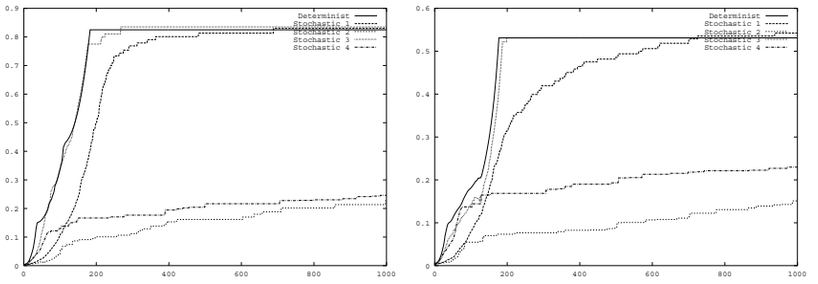


Fig. 2. Synthetic Data set with 10% noise (left) and 30% noise (right)

The results obtained are similar to those found in the perfect case. The convergence speeds are in the same order.

The previous graphics mask an important point: the required time for each step. The table 1 gathers the computation time, expressed in seconds, used for 10000 iterations. For information, they are obtained on a Pentium II 300Mhz with 32 Mb memory.

Table 1. Computation time (in second) used by the several algorithms on different data sets

	Perfect	10% noise	30 % noise
detreminite	204	250	275
Stochastic 1	2.66	4	4
Stochastic 2	23	31	31
Stochastic 3	39	130	130
Stochastic 4	81	110	120

The deterministic procedure is very high time consuming. The first stochastic procedure seems to be a good compromise between accuracy and time consumption.

Then we apply the algorithms on a well known benchmark: 1984 United States Congressional Voting Records Database. We remove the attribute expressing the vote. On Fig.3 (left) is plotted the $\tau_{\mathcal{Q}}$ values for each iteration of the algorithms. On the contrary of the previous experiences, the distinction between on one hand the second and the fourth stochastic procedures, and on the other hand the other procedures, is less obvious. All the procedures find quite the same partition. We can observe an unexpected decrease in the function. This is due to the fact that an increase on $\tau_{\mathcal{O}}$ leads to a decrease on $\tau_{\mathcal{Q}}$. Such phenomenon appears rarely, and when it appears, the algorithm quickly restores a better partition. Consequently this is not an handicap in the optimization process. To visualize the influence of the $\tau_{\mathcal{O}}$ optimization on the $\tau_{\mathcal{V}}$ one, we plot (see Fig.3 (right)) the value of the both functions at each iterations. On this graph we clearly observe the compensation process on the optimization of both functions.

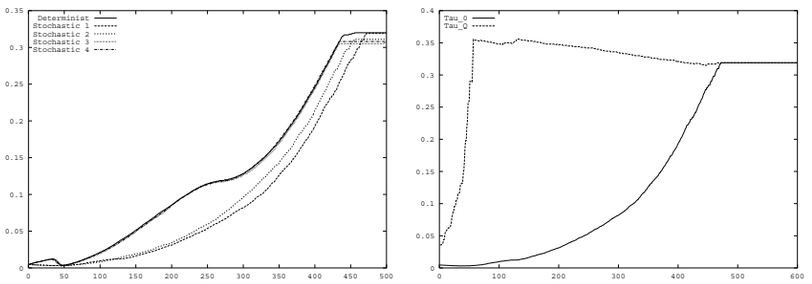


Fig. 3. Vote Data Set (left), Values of $\tau_{\mathcal{O}}$ and $\tau_{\mathcal{Q}}$ when using Stochastic 1 (right)

The quality of the obtained partition of voters can be evaluated through its comparison with the results of the elections. This election consists in deciding between a democrat or republican congress. We also obtain a partition in two clusters. The table 2 crosses the two partitions.

Table 2. Cross table of the votes and the obtained results by the algorithm 4

Our results vs Vote	Democrat	Republican	♯
P_1	221	14	235
P_2	46	154	200
♯	267	168	435

The group P_1 of the obtained partition seems obviously corresponds to the democrat one, whereas the group P_2 fits the republican population. The rate of accurate prediction is here of 86.2% whereas about 90% accuracy appears to be STAGGER's asymptote.

The quality of the partition on the set of attribute-value pairs is also very good. We denote by G_1 the cluster of attribute-value pairs associated to the group P_1, which fits well the democrat population. This set gathers all the attribute-value pairs whose conditional probability of appearance, given the fact the voter is Democrat, are superior of the ones associated to the Republican voters. The probability of being democrat knowing the voter owns this attribute-value pair is also superior of the one obtained for the republican voters and thus for all the attribute-value pairs. All attributes are of binary/type (yes/no), and for each attribute, the yes value belong to a cluster and the no one to another one. Consequently, we can say that the obtained partition is ideal regarding our criteria of a good partition.

7 Conclusion

In this article, we have presented a variational study of a function used for guiding the search of a partition in conceptual clustering. It consists in evaluating the variation of the function when transfer, merge or split operators are applied to modify a partition. We showed that using this approach in optimization procedure allows to decrease the computational cost. Furthermore, it leads to simplify the problem, expressing the three operators under a single one.

We mix this approach with stochastic local search optimization procedures and apply them on a synthetic data set and the real data set *Vote* taken from the UCI Irvine repository. The experimentation leads to conclude that some randomness is needed in the local search procedure to speed up the convergence to the best partition. But too much randomness, when the procedure examine a random subset of attribute-value pairs of a cluster, slow down the convergence in a more important way. The partitions obtained on the *Vote* data set are both of excellent accuracy. The partition on the voters set is quite the same than the one given by the result of the election, without taking this information into account. The partition on the set of attribute-value pairs follows exactly the conditional probabilities of appearance of those attribute-value pairs given the vote class.

In a future work, we plan to analytically approximate the combination of (λ_i^y) which most improve the quality of the partition. This would reduce the number of steps of optimization required to obtain an optimum.

References

[BRE91] J. N. Bhuyan, V. V. Raghavan, and V. K. Elayavalli. Genetic algorithm for clustering with ordered representation. In Richard K. Belew and Lashon B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[CDG+88] G. Celeux, E. Diday, G. Govaert, Y. Lechevallier, and H. Ralambondrainy. *Classification automatique des données*. Dunod, paris, 1988.

[Col98] R. M. Cole. Clustering with genetic algorithms. Master's thesis, University of Western Australia, 1998.

[CS96] P. Cheeseman and J. Stutz. Bayesian classification (autoclass): Theory and results. *Advances in Knowledge Discovery and Data Mining*, 1996.

[Fis87] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[Fis96] D. H. Fisher. Iterative optimization and simplification of hierarchical clusterings. *Journal of Artificial Intelligence Research*, 4:147–180, 1996.

[FK00] P. Fränti and J. Kivijärvi. Randomised local search algorithm for the clustering problem. *Pattern Analysis and Applications*, pages 358–369, 2000.

[GK54] L. A. Goodman and W. H. Kruskal. Measures of association for cross classification. *Journal of the American Statistical Association*, 49:732–764, 1954.

[GKLN00] M. Gyllenberg, T. Koski, T. Lund, and O. Nevalainen. Clustering by adaptive local search with multiple search operators. *Pattern Analysis and Applications*, pages 348–357, 2000.

[Gov84] G. Govaert. Classification simultanée de tableaux binaires. In E. Diday, M. Jambu, L. Lebart, J. Pages, and R. Tomassone, editors, *Data analysis and informatics III*, pages 233–236. North Holland, 1984.

[JD88] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice Hall, Englewood cliffs, New Jersey, 1988.

[LdC96] I.C. Lerman and J. F. P. da Costa. Coefficients d'association et variables à très grand nombre de catégories dans les arbres de décision : application à l'identification de la structure secondaire d'une protéine. Technical Report 2803, INRIA, février 1996.

[MH91] G. Matthews and J. Hearne. Clustering without a metric. *IEEE Transaction on pattern analysis and machine intelligence*, 13(2):175–184, 1991.

[MSW72] W. T. McCormick, P. J. Schweitzer, and T. W. White. Problem decomposition and data reorganization by a clustering technique. *Operations Research*, 20(5):993–1009, 1972.

[Ols95] M. Olszak. *Modélisation des relations de causalité entre variables qualitatives*. PhD thesis, Université de Genève, 1995.

[Rak97] R. Rakotomalala. *Graphes d'induction*. PhD thesis, Université Claude Bernard Lyon 1, 1997.

[RF01] C. Robardet and F. Feschet. Comparison of three objective functions for conceptual clustering. In *Proceedings of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases*. Springer-Verlag, September 2001.

[Rud94] G. Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on neuronal networks*, 5(1):96–101, 1994.

[SCH75] J.R. Slagle, C. L. Chang, and S. R. Heller. A clustering and data-reorganizing. *IEEE Transactions On systems, Man and Cybernetics*, pages 125–128, January 1975.

Computational Revision of Quantitative Scientific Models

Kazumi Saito[1], Pat Langley[2], Trond Grenager[2],
Christopher Potter[3], Alicia Torregrosa[3], and Steven A. Klooster[3]

[1] NTT Communication Science Laboratories
2-4 Hikaridai, Seika, Soraku, Kyoto 619-0237 Japan
saito@cslab.kecl.ntt.co.jp

[2] Computational Learning Laboratory, CSLI
Stanford University, Stanford, California 94305 USA
{langley,grenager}@cs.stanford.edu

[3] Ecosystem Science and Technology Branch
NASA Ames Research Center, MS 242-4
Moffett Field, California 94035 USA
{cpotter,lisy,sklooster}@gaia.arc.nasa.gov

Abstract. Research on the computational discovery of numeric equations has focused on constructing laws from scratch, whereas work on theory revision has emphasized qualitative knowledge. In this paper, we describe an approach to improving scientific models that are cast as sets of equations. We review one such model for aspects of the Earth ecosystem, then recount its application to revising parameter values, intrinsic properties, and functional forms, in each case achieving reduction in error on Earth science data while retaining the communicability of the original model. After this, we consider earlier work on computational scientific discovery and theory revision, then close with suggestions for future research on this topic.

1 Research Goals and Motivation

Research on computational approaches to scientific knowledge discovery has a long history in artificial intelligence, dating back over two decades (e.g., Langley, 1979; Lenat, 1977). This body of work has led steadily to more powerful methods and, in recent years, to new discoveries deemed worth publication in the scientific literature, as reviewed by Langley (1998). However, despite this progress, mainstream work on the topic retains some important limitations.

One drawback is that few approaches to the intelligent analysis of scientific data can use available knowledge about the domain to constrain search for laws or explanations. Moreover, although early work on computational discovery cast discovered knowledge in notations familiar to scientists, more recent efforts have not. Rather, influenced by the success of machine learning and data mining, many researchers have adopted formalisms developed by these fields, such as decision trees and Bayesian networks. A return to methods that operate on established scientific notations seems necessary for scientists to understand their results.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 336–349, 2001.

Like earlier research on computational scientific discovery, our general approach involves defining a space of possible models stated in an established scientific formalism, specifically sets of numeric equations, and developing techniques to search that space. However, it differs from previous work in this area by starting from an existing scientific model and using heuristic search to revise the model in ways that improve its fit to observations. Although there exists some research on theory refinement (e.g., Ourston & Mooney 1990; Towell, 1991), it has emphasized qualitative knowledge rather than quantitative models that relate continuous variables, which play a central role in many sciences.

In the pages that follow, we describe an approach to revising quantitative models of complex systems. We believe that our approach is a general one appropriate for many scientific domains, but we have focused our efforts on one area – certain aspects of the Earth ecosystem – for which we have a viable model, existing data, and domain expertise. We briefly review the domain and model before moving on to describe our approach to knowledge discovery and model revision. After this, we present some initial results that suggest our approach can improve substantially the model's fit to available data. We close with a discussion of related discovery work and directions for future research.

2 A Quantitative Model of the Earth Ecosystem

Data from the latest generation of satellites, combined with readings from ground sources, hold great promise for testing and improving existing scientific models of the Earth's biosphere. One such model, CASA, developed by Potter and Klooster (1997, 1998) at NASA Ames Research Center, accounts for the global production and absorption of biogenic trace gases in the Earth atmosphere, as well as predicting changes in the geographic patterns of major vegetation types (e.g., grasslands, forest, tundra, and desert) on the land.

CASA predicts, with reasonable accuracy, annual global fluxes in trace gas production as a function of surface temperature, moisture levels, and soil properties, together with global satellite observations of the land surface. The model incorporates difference equations that represent the terrestrial carbon cycle, as well as processes that mineralize nitrogen and control vegetation type. These equations describe relations among quantitative variables and lead to changes in the modeled outputs over time. Some processes are contingent on the values of discrete variables, such as soil type and vegetation, which take on different values at different locations. CASA operates on gridded input at different levels of resolution, but typical usage involves grid cells that are eight kilometers square, which matches the resolution for satellite observations of the land surface.

To run the CASA model, the difference equations are repeatedly applied to each grid cell independently to produce new variable values on a daily or monthly basis, leading to predictions about how each variable changes, at each location, over time. Although CASA has been quite successful at modeling Earth's ecosystem, there remain ways in which its predictions differ from observations, suggesting that we invoke computational discovery methods to improve its ability to fit the data. The result would be a revised model, cast in the same notation as the

Table 1. Variables used in the NPPc portion of the CASA ecosystem model.

NPPc is the net plant production of carbon at a site during the year.
E is the photosynthetic efficiency at a site after factoring various sources of stress.
T1 is a temperature stress factor ($0 < T1 < 1$) for cold weather.
T2 is a temperature stress factor ($0 < T2 < 1$), nearly Gaussian in form but falling off more quickly at higher temperatures.
W is a water stress factor ($0.5 < W < 1$) for dry regions.
Topt is the average temperature for the month at which MON-FAS-NDVI takes on its maximum value at a site.
Tempc is the average temperature at a site for a given month.
EET is the estimated evapotranspiration (water loss due to evaporation and transpiration) at a site.
PET is the potential evapotranspiration (water loss due to evaporation and transpiration given an unlimited water supply) at a site.
PET-TW-M is a component of potential evapotranspiration that takes into account the latitude, time of year, and days in the month.
A is a polynomial function of the annual heat index at a site.
AHI is the annual heat index for a given site.
MON-FAS-NDVI is the relative vegetation greenness for a given month as measured from space.
IPAR is the energy from the sun that is intercepted by vegetation after factoring in time of year and days in the month.
FPAR-FAS is the fraction of energy intercepted from the sun that is absorbed photosynthetically after factoring in vegetation type.
MONTHLY-SOLAR is the average solar irradiance for a given month at a site.
SOL-CONVER is 0.0864 times the number of days in each month.
UMD-VEG is the type of ground cover (vegetation) at a site.

original one, that incorporates changes which are interesting to Earth scientists and which improve our understanding of the environment.

Because the overall CASA model is quite complex, involving many variables and equations, we decided to focus on one portion that lies on the model's 'fringes' and that does not involve any difference equations. Table 1 describes the variables that occur in this submodel, in which the dependent variable, NPPc, represents the net production of carbon. As Table 2 indicates, the model predicts this quantity as the product of two unobservable variables, the photosynthetic efficiency, E, at a site and the solar energy intercepted, IPAR, at that site.

Photosynthetic efficiency is in turn calculated as the product of the maximum efficiency (0.56) and three stress factors that reduce this efficiency. One stress term, T2, takes into account the difference between the optimum temperature, Topt, and actual temperature, Tempc, for a site. A second factor, T1, involves

Table 2. Equations used in the NPPc portion of the CASA ecosystem model.

$$\text{NPPc} = \sum_{month} \max(\text{E} \cdot \text{IPAR}, 0)$$
$$\text{E} = 0.56 \cdot \text{T1} \cdot \text{T2} \cdot \text{W}$$
$$\text{T1} = 0.8 + 0.02 \cdot \text{Topt} - 0.0005 \cdot \text{Topt}^2$$
$$\text{T2} = 1.18/[(1 + e^{0.2 \cdot (\text{Topt} - \text{Tempc} - 10)}) \cdot (1 + e^{0.3 \cdot (\text{Tempc} - \text{Topt} - 10)})]$$
$$\text{W} = 0.5 + 0.5 \cdot \text{EET/PET}$$
$$\text{PET} = 1.6 \cdot (10 \cdot \text{Tempc} \ / \ \text{AHI})^{A} \cdot \text{PET-TW-M if Tempc} > 0$$
$$\text{PET} = 0 \text{ if Tempc} \leq 0$$
$$\text{A} = 0.000000675 \cdot \text{AHI}^3 - 0.0000771 \cdot \text{AHI}^2 + 0.01792 \cdot \text{AHI} + 0.49239$$
$$\text{IPAR} = 0.5 \cdot \text{FPAR-FAS} \cdot \text{MONTHLY-SOLAR} \cdot \text{SOL-CONVER}$$
$$\text{FPAR-FAS} = \min((\text{SR-FAS} - 1.08)/\text{SRDIFF(UMD-VEG)}, 0.95)$$
$$\text{SR-FAS} = -(\text{MON-FAS-NDVI} + 1000) \ / \ (\text{MON-FAS-NDVI} - 1000)$$

the nearness of Topt to a global optimum for all sites, reflecting the intuition that plants which are better adapted to harsh temperatures are less efficient overall. The third term, W, represents stress that results from lack of moisture as reflected by EET, the estimated water loss due to evaporation and transpiration, and PET, the water loss due to these processes given an unlimited water supply. In turn, PET is defined in terms of the annual heat index, AHI, for a site, and PET-TW-M, another component of potential evapotranspiration.

The energy intercepted from the sun, IPAR, is computed as the product of FPAR-FAS, the fraction of energy absorbed photosynthetically for a given vegetation type, MONTHLY-SOLAR, the average radiation for a given month, and SOL-CONVER, the number of days in that month. FPAR-FAS is a function of MON-FAS-NDVI, which indicates relative greenness at a site as observed from space, and SRDIFF, an intrinsic property that takes on different numeric values for different vegetation types as specified by the discrete variable UMD-VEG.

Of the variables we have mentioned, NPPc, Tempc, MONTHLY-SOLAR, SOL-CONVER, MON-FAS-NDVI, and UMD-VEG are observable. Three additional terms – EET, PET-TW-M, and AHI – are defined elsewhere in the model, but we assume their definitions are correct and thus we can treat them as observables. The remaining variables are unobservable and must be computed from the others using their definitions. This portion of the model also contains a number of numeric parameters, as shown in the equations in Table 2.

3 An Approach to Quantitative Model Revision

As noted earlier, our approach to scientific discovery involves refining models like CASA that involve relations among quantitative variables. We adopt the traditional view of discovery as heuristic search through a space of models, with the search process directed by candidates' ability to fit the data. However, we assume this process starts not from scratch, but rather with an existing model,

and the search operators involve making changes to this model, rather than constructing entirely new structures.

Our long-term goal is not to automate the revision process, but instead to provide an interactive tool that scientists can direct and use to aid their model development. As a result, the approach we describe in this section addresses the task of making local changes to a model rather than carrying out global optimization, as assumed by Chown and Dietterich (2000). Thus, our software takes as input not only observations about measurable variables and an existing model stated as equations, but also information about which portion of the model should be altered. The output is a revised model that fits the observed data better than the initial one.

Below we review two discovery algorithms that we utilize to improve the specified part of a model, then describe three distinct types of revision they support. We consider these in order of increasing complexity, starting with simple changes to parameter values, moving on to revisions in the values of intrinsic properties, and ending with changes in an equation's functional form.

3.1 The RF5 and RF6 Discovery Algorithms

Our approach relies on RF5 and RF6, two algorithms for discovering numeric equations described Saito and Nakano (1997, 2000). Given data for some continuous variable y that is dependent on continuous predictive variables $x_1, \ldots, x_n$, the RF5 system searches for multivariate polynomial equations of the form

$$y \;=\; w_0 + \sum_{j=1}^{J} w_j \prod_{k=1}^{K} x_k^{w_{jk}} \;=\; w_0 + \sum_{j=1}^{J} w_j \exp\left(\sum_{k=1}^{K} w_{jk} \ln(x_k)\right), \qquad (1)$$

Such functional relations subsume many of the numeric laws found by previous computational discovery systems like Bacon (Langley, 1979) and Fahrenheit (Żytkow, Zhu, & Hussam, 1990).

RF5's first step involves transforming a candidate functional form with J summed terms into a three-layer neural network based on the rightmost form of expression (1), in which the K hidden nodes in this network correspond to *product units* (Durbin & Rumelhart, 1989). The system then carries out search through the weight space using the BPQ algorithm, a second-order learning technique that calculates both the descent direction and the step size automatically.

This process halts when it finds a set of weights that minimize the squared error on the dependent variable y. RF5 runs the BPQ method on networks with different numbers of hidden units, then selects the one that gives the best score on an MDL metric. Finally, the program transforms the resulting network into a polynomial equation, with weights on hidden units becoming exponents and other weights becoming coefficients.

The RF6 algorithm extends RF5 by adding the ability to find conditions on a numeric equation that involve nominal variables, which it encodes using one input variable for each nominal value. To this end, the system first generates one such condition for each training case, then utilizes k-means clustering to generate

a smaller set of more general conditions, with the number of clusters determined through cross validation. Finally, RF6 invokes decision-tree induction to construct a classifier that discriminates among these clusters, which it transforms into rules that form the nominal conditions on the polynomial equation that RF5 has generated.

3.2 Three Types of Model Refinement

There exist three natural types of refinement within the class of models, like CASA, that are stated as sets of equations that refer to unobservable variables. These include revising the parameter values in equations, altering the values for an intrinsic property, and changing the functional form of an existing equation.

Improving the parameters for an equation is the most straightforward process. The NPPc portion of CASA contains some parameterized equations that our Earth science team members believe are reliable, like that for computing the variable A from AHI, the annual heat index. However, it also includes equations with parameters about which there is less certainty, like the expression that predicts the temperature stress factor T2 from Tempc and Topt. Our approach to revising such parameters relies on creating a specialized neural network that encodes the equation's functional form using ideas from RF5, but also including a term for the unchanged portion of the model. We then run the BPQ algorithm to find revised parameter values, initializing weights based on those in the model.

We can utilize a similar scheme to improve the values for an intrinsic property like SRDIFF that the model associates with the discrete values for some nominal variable like UMD-VEG (vegetation type). We encode each nominal term as a set of dummy variables, one for each discrete value, making the dummy variable equal to one if the discrete value occurs and zero otherwise. We introduce one hidden unit for the intrinsic property, with links from each of the dummy variables and with weights that correspond to the intrinsic values associated with each discrete value. To revise these weights, we create a neural network that incorporates the intrinsic values but also includes a term for the unchanging parts of the model. We can then run BPQ to revise the weights that correspond to intrinsic values, again initializing them to those in the initial model.

Altering the form of an existing equation requires somewhat more effort, but maps more directly onto previous work in equation discovery. In this case, the details depend on the specific functional form that we provide, but because we have available the RF5 and RF6 algorithms, the approach supports any of the forms that they can discover or specializations of them. Again, having identified a particular equation that we want to improve, we create a neural network that encodes the desired form, then invoke the BPQ algorithm to determine its parametric values, in this case initializing the network weights randomly.

This approach to model refinement supports changes to only one equation or intrinsic property at a time, but this is consistent with the interactive process described earlier. We envision the scientist identifying a portion of the model that he thinks could be better, running one of the three revision methods to improve its fit to the data, and repeating this process until he is satisfied.

4 Initial Results on Ecosystem Data

In order to evaluate our approach to scientific model revision, we utilized data relevant to the NPPc model available to the Earth science members of our team. These data consisted of observations from 303 distinct sites with known vegetation type and for which measurements of Tempc, MON-FAS-NDVI, MONTHLY-SOLAR, SOL-CONVER, and UMD-VEG were available for each month during the year. In addition, other portions of CASA were able to compute values for the variables AHI, EET, and PET-TW-M. The resulting 303 training cases seemed sufficient for initial tests of our revision methods, so we used them to drive a variety of changes to the handcrafted model of carbon production.

4.1 Results on Parameter Revision

Our Earth science team members identified the equation for T2, one of the temperature stress variables, as a likely candidate for revision. As noted earlier, the handcrafted expression for this term was

$$T2 = 1.8/[(1 + e^{0.2(Topt - Tempc - 10)})(1 + e^{-0.3(Tempc - Topt - 10)})] \ ,$$

which produces a Gaussian-like curve that is slightly assymetrical. This reflects the intuition that photosynthetic efficiency will decrease when temperature (Tempc) is either below or above the optimal (Topt).

To improve upon this equation, we defined $x = \text{Topt} - \text{Tempc}$ as an intermediate variable and recast the expression for T2 as the product of two sigmoidal functions of the form $\sigma(a) = 1/(1 + \exp(-a))$ and a parameter. We transformed these into a neural network and used BPQ to minimize the error function

$$\mathcal{F}_1 = \sum_{sample} (\text{NPPc} - \sum_{month} w_0 \cdot \sigma(v_{10} + v_{11} \cdot x) \cdot \sigma(v_{20} - v_{21} \cdot x) \cdot \text{Rest})^2 \ ,$$

over the parameters $\{w_0, v_{10}, v_{11}, v_{20}, v_{21}\}$, where $\text{Rest} = 0.56 \cdot \text{T1} \cdot \text{W} \cdot \text{IPAR}$. The resulting equation generated in this manner was

$$T2 = 1.80/[(1 + e^{0.05(Topt - Tempc - 10.8}})(1 + e^{-0.03(Tempc - Topt - 90.33}})] \ ,$$

which has reasonably similar values to the original ones for some parameters but quite different values for others.

The root mean squared error (RMSE) for the original model on the available data was 467.910. In contrast, the error for the revised model was 457.757 on the training data and 461.466 using leave-one-out cross validation. Thus, RF6's modification of parameters in the T2 equation produced slightly more than one percent reduction in overall model error, which is somewhat disappointing.

However, inspection of the resulting curves reveals a more interesting picture. Plotting the temperature stress factor T2 using the revised equations as a function of the difference Topt − Tempc still gives a Gaussian-like curve, but within the effective range (from −30 to 30 Celsius) its values decrease monotonically. This seems counterintuitive but interesting from an Earth science perspective,

as it suggests this stress factor has little influence on NPPc. Moreover, the original equation for T2 was not well grounded in first principles of plant physiology, making empirical improvements of this sort beneficial to the modeling enterprise.

As another candidate for parameter revision, we selected the PET equation,

$$\text{PET} = 1.6 \cdot (10 \cdot \max(\text{Tempc}, 0) \ / \ \text{AHI})^{A} \cdot \text{PET-TW-M} \ ,$$

which calculates potential water loss due to evaporation and transpiration given an unlimited water supply. By transforming this expression into

$$\text{PET} = \exp(\ln(1.6) + \text{A} \cdot \ln(10)) \cdot (\max(\text{Tempc}, 0) \ / \ \text{AHI})^{A} \cdot \text{PET-TW-M}$$

and replacing the parameter values ln(1.6) and ln(10) with the variables v_0 and v_1, we constructed a neural network and used BPQ for error minimization. When transforming the trained network back into the original form, the equation that resulted was

$$\text{PET} = 1.56 \cdot (9.16 \cdot \max(\text{Tempc}, 0) \ / \ \text{AHI})^{A} \cdot \text{PET-TW-M} \ ,$$

which has values that are very similar to those in the original model's equation.

Moreover, since the RMSE for the obtained model was 464.358 on the training data and 467.643 using leave-one-out cross validation, the revision process did not improve the model's accuracy substantially. However, since the PET equation is based on Thornthwaite's (1948) method, which has been used continuously for over 50 years, we should not be overly surprised at this negative result. Indeed, we are encouraged by the fact that our approach did not revise parameters that have stood the test of time in Earth science.

4.2 Results on Intrinsic Value Revision

Another portion of the NPPc model that held potential for revision concerns the intrinsic property SRDIFF associated with the vegetation type UMD-VEG. For each site, the latter variable takes on one of 11 nominal values, such as grasslands, forest, tundra, and desert, each with an associated numeric value for SRDIFF that plays a role in the FPAR-FAS equation. This gives 11 parameters to revise, which seems manageable given the number of observations available.

As outlined earlier, to revise these intrinsic values, we introduced one dummy variable, UMD-VEG_k, for each vegetation type such that $\text{UMD-VEG}_k = 1$ if $\text{UMD-VEG} = k$ and 0 otherwise. We then defined SRDIFF(UMD-VEG) as $\exp(-\sum_k v_k \cdot \text{UMD-VEG}_k)$ and, since SRDIFF's value is independent of the month, we used BPQ to minimize, over the weights $\{v_k\}$, the error function

$$\mathcal{F}_2 = \sum_{site} \left(\text{NPPc} - \exp(\textstyle\sum_k v_k \cdot \text{UMD-VEG}_k) \cdot \text{Rest}\right)^2 \ ,$$

where $\text{Rest} = \sum_{month} E \cdot 0.5 \cdot (\text{SR-FAS} - 1.08) \cdot \text{MONTHLY-SOLAR} \cdot \text{SOL-CONVER}$.

Table 3 shows the initial values for this intrinsic property, as set by the CASA developers, along with the revised values produced by the above approach when

Table 3. Original and revised values for the SRDIFF intrinsic property, along with the frequency for each vegetation type.

vegetation type	A	B	C	D	E	F	G	H	I	J	K
original	3.06	4.35	4.35	4.05	5.09	3.06	4.05	4.05	4.05	5.09	4.05
revised	2.57	4.77	2.20	3.99	3.70	3.46	2.34	0.34	2.72	3.46	1.60
clustered	2.42	3.75	2.42	3.75	3.75	3.75	2.42	0.34	2.42	3.75	2.42
frequency	3.3	8.9	0.3	3.6	21.1	19.1	15.2	3.3	19.1	2.3	3.6

we fixed other parts of the NPPc model. The most striking result is that the revised intrinsic values are nearly always lower than the initial values. The RMSE for the original model was 467.910, whereas the error using the revised values was 432.410 on the training set and 448.376 using cross validation. The latter constitutes an error reduction of over four percent, which seems substantial.

However, since the original 11 intrinsic values were grouped into only four distinct values, we applied RF6's clustering procedure over the trained neural network to group the revised values in the same manner. We examined the effect on error rate as we varied the number of clusters from one to five; as expected, the training RMSE decreased monotonically, but the cross-validation RMSE was minimized for three clusters of values. The estimated error for this revised model is slightly better than for the one with 11 distinct values.

Again, the clustered values are nearly always lower than the initial ones, a result that is certainly interesting from an Earth science viewpoint. We suspect that measurements of NPPc and related variables from a wider range of sites would produce intrinsic values closer to those in the original model. However, such a test must await additional observations and, for now, empirical fit to the available data should outweigh the theoretical basis for the initial settings.

In another approach to revising intrinsic values, we retained the original grouping of vegetation types into sets, with each type in a given set having the same value. We utilized a weight-sharing technique to encode this background knowledge in a neural network. For example, let v_A and v_F be weights corresponding to the SRDIFF values for vegetation types A and F, respectively; to ensure these values remained the same, we treated them as a single weight, say v_{AF}. Here we can see that BPQ calculates the derivative of the error function over v_{AF} as a sum of the individual derivatives over v_A and v_F,

$$\frac{\partial \mathcal{F}_2}{\partial v_{AF}} = \frac{\partial \mathcal{F}_2}{\partial v_A} + \frac{\partial \mathcal{F}_2}{\partial v_F} \ .$$

In the trained neural network, the derivative over v_{AF} becomes zero, but there is no guarantee that each derivative over v_A or v_F will do so. Therefore, we can treat the sum of the absolute values for derivatives over shared weights, like v_A and v_F, as a criterion for the 'unlikeness' among the elements of such a grouping.

Table 4 shows the revised values for the intrinsic property SRDIFF that result from this approach, along with values for the unlikeness criterion defined above.

Table 4. Original and revised values, using the original groupings, for the SRDIFF intrinsic property, along with the frequency and unlikeness for each vegetation group.

vegetation type	A∨F	B∨C	E∨J	D∨G∨H∨I∨K
original	3.06	4.35	5.09	4.05
revised	2.23	3.27	2.54	1.81
frequency	22.4	9.2	23.4	44.9
unlikeness	26.1	0.3	2.3	13.6

As before, the obtained intrinsic values are always lower than the initial ones, and our criterion suggests that the group containing the vegetation types A and F has the least coherence. The RMSE for the revised model was 442.782 on the training data and 449.097 using leave-one-out cross validation, again indicating about four percent reduction in the model's overall error.

4.3 Results on Revising Equation Structure

We also wanted to demonstrate our approach's ability to improve the functional form of the NPPc model. For this purpose, we selected the equation for photosynthetic efficiency,

$$E = 0.56 \cdot T1 \cdot T2 \cdot W \ ,$$

which states that this term is a product of the water stress term, W, and the two temperature stress terms, T1 and T2. Because each stress factor takes on values less than one, multiplication has the effect of reducing photosynthetic efficiency E below the maximum 0.56 possible (Potter & Klooster, 1998).

Since E is calculated as a simple product of the three variables, one natural extension was to consider an equation that included exponents on these terms. To this end, we borrowed techniques from the RF5 system to create a neural network for such an expression, then used BPQ to minimize the error function

$$\mathcal{F}_3 = \sum_{site} \left(\mathrm{NPPc} - \sum_{month} u_0 \cdot \mathrm{T1}^{u_1} \cdot \mathrm{T2}^{u_2} \cdot \mathrm{W}^{u_3} \cdot \mathrm{IPAR}\right)^2 \ ,$$

over the parameters $\{u_0, u_1, u_2, u_3\}$, which assumes the equations that predict IPAR remain unchanged. We initialized u_0 to 0.56 and the other parameters to 1.0, as in the original model, and constrained the latter to be positive. The revised equation found in this manner,

$$E = 0.521 \cdot T1^{0.00} \cdot T2^{0.03} \cdot W^{0.00} \ ,$$

has a small exponent for T2 and zero exponents for T1 and W, suggesting the former influences photosynthetic efficiency in minor ways and the latter not at all. On the available data, the root mean squared error for the original model was 467.910. In contrast, the revised model has an RMSE of 443.307 on the training set and an RMSE of 446.270 using cross validation. Thus, the revised

equation produces a substantially better fit to the observations than does the original model, in this case reducing error by almost five percent.

With regards to Earth science, these results are plausible and the most interesting of all, as they suggest that the T1 and W stress terms are unnecessary for predicting NPPc. One explanation is that the influence of these factors is already being captured by the NDVI measure available from space, for which the signal-to-noise ratio has been steadily improving since CASA was first developed.

These results encouraged us to explore more radical revisions to the functional form for photosynthetic efficiency. Thus, we told our system to consider a form that omitted the three stress factors but that included the four variables – Topt, Tempc, EET, and PET – that appear in their definitions:

$$\mathrm{E} = v_0 \cdot \exp(-0.5 \cdot (v_1 \cdot \mathrm{Topt} + v_2 \cdot \mathrm{Tempc} + v_3 \cdot \mathrm{EET} + v_4 \cdot \mathrm{PET} + v_5)^2) \ .$$

This Gaussian-like activation function satisfies the constraint that E is positive and less than one. Running BPQ to minimize the error function over $\{v_0, \ldots v_5\}$ produced the equation

$$\mathrm{E} = 0.57 \cdot \exp(-0.5 \cdot (-0.04 \cdot \mathrm{Topt} + 0.03 \cdot \mathrm{Tempc} - 0.03 \cdot \mathrm{EET} + 0.01 \cdot \mathrm{PET})^2),$$

where we eliminated the parameter v_5 because its value was -0.003. The RMSE for the revised model was 439.101 on the training data and 444.470 using leave-one-out cross validation, indicating more than five percent reduction in error.

These results are very similar to those from our first approach, which produced a cross validation RMSE of 446.270. In this case, the revised model is simpler in that it defines E directly in terms of Topt, Tempc, EET, and PET, rather than relying on the theoretical terms T1, T2, and W, two of which provide no predictive power. On the other hand, the original form for E had a clear theoretical interpretation, whereas the new version does not. In such situations, the final decision should be left to domain scientists, who are best suited to balance a model's simplicity against its interepretability.

5 Related Research on Computational Discovery

Our research on computational scientific discovery draws on two previous lines of work. One approach, which has an extended history within artificial intelligence, addresses the discovery of explicit quantitative laws. Early systems for numeric law discovery like Bacon (Langley, 1979; Langley et al., 1987) carried out a heuristic search through a space of new terms and simple equations. Numerous successors like Fahrenheit (Żytkow et al., 1990) and RF5 (Saito & Nakano, 1997) incorporate more sophisticated and more extensive search through a larger space of numeric equations.

The most relevant equation discovery systems take into account domain knowledge to constrain the search for numeric laws. For example, Kokar's (1986) Coper utilized knowledge about the dimensions of variables to focus attention and, more recently, Washio and Motoda's (1998) SDS extends this idea to support different types of variables and sets of simultaneous equations. Todorovski

and Džeroski's (1997) LaGramge takes a quite different approach, using domain knowledge in the form of context-free grammars to constrain its search through a space of differential equation models that describe temporal behavior.

Although research on computational discovery of numeric laws has emphasized communicable scientific notations, it has focused on constructing such laws rather than revising existing ones. In contrast, another line of research has addressed the refinement of existing models to improve their fit to observations. For example, Ourston and Mooney (1990) developed a method that used training data to revise models stated as sets of propositional Horn clauses. Towell (1991) reports another approach that transforms such models into multilayer neural networks, then uses backpropagation to improve their fit to observations, much as we have done for numeric equations. Work in this paradigm has emphasized classification rather than regression tasks, but one can view our work as adapting the basic approach to equation discovery.

We should also mention related work on the automated improvement of ecosystem models. Most AI work on Earth science domains focuses on learning classifiers that predict vegetation from satellite measures like NDVI, as contrasted with our concern for numeric prediction. Chown and Dietterich (2000) describe an approach that improves an existing ecosystem model's fit to continuous data, but their method only alters parameter values and does not revise equation structure. On another front, Schwabacher and Langley (2001) use a rule-induction algorithm to discover piecewise linear models that predict NDVI from climate variables, but their method takes no advantage of existing models.

6 Directions for Future Research

Although we have been encouraged by our results to date, there remain a number of directions in which we must extend our approach before it can become a useful tool for scientists. As noted earlier, we envision an interactive discovery aide that lets the user focus the system's attention on those portions of the model it should attempt to improve. To this end, we need a graphical interface that supports marking of parameters, intrinsic properties, and equations that can be revised, as well as tools for displaying errors as a function of space, time, and predictive variables.

In addition, the current system is limited to revising the parameters or form of one equation in the model at a time, as well as requiring some handcrafting to encode the equations as a neural network. Future versions should support revisions of multiple equations at the same time, preferably invoking the same variants of backpropagation as we have used to date, and also provide a library that maps functional forms to neural network encodings, so the system can transform the former into the latter automatically. We should also explore using other approaches to equation discovery, such as Todorovski and Džeroski's LaGramge, in place of the RF6 algorithm.

Naturally, we also hope to evaluate our approach on its ability to improve other portions of the CASA model, as additional data becomes available. Another test of generality would be application of the same methods to other sci-

entific domains in which there already exist formal models that can be revised. In the longer term, we should evaluate our interactive system not only in its ability to increase the predictive accuracy of an existing model, but in terms of the satisfaction to scientists who use the system to that end.

Another challenge that we have encountered in our research has been the need to translate the existing CASA model into a declarative form that our discovery system can manipulate. In response, another long-term goal involves developing a modeling language in which scientists can cast their initial models and carry out simulations, but that can also serve as the declarative representation for our discovery methods. The ability to automatically revise models places novel constraints on such a language, but we are confident that the result will prove a useful aid to the discovery process.

7 Concluding Remarks

In this paper, we addressed the computational task of improving an existing scientific model that is composed of numeric equations. We illustrated this problem with an example model from the Earth sciences that predicts carbon production as a function of temperature, sunlight, and other variables. We identified three activities that can improve a model – revising an equation's parameters, altering the values of an intrinsic property, and changing the functional form of an equation, then presented results for each type on an ecosystem modeling task that reduced the model's prediction error, sometimes substantially.

Our research on model revision builds on previous work in numeric law discovery and qualitative theory refinement, but it combines these two themes in novel ways to enable new capabilities. Clearly, we remain some distance from our goal of an interactive discovery tool that scientists can use to improve their models, but we have also taken some important steps along the path, and we are encouraged by our initial results on an important scientific problem.

References

Chown, E., & Dietterich, T. G. (2000). A divide and conquer approach to learning from prior knowledge. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 143–150). San Francisco: Morgan Kaufmann.

Durbin, R. & Rumelhart, D. E. (1989). Product units: A computationally powerful and biologically plausible extension. *Neural Computation*, *1*, 133–142.

Kokar, M. M. (1986). Determining arguments of invariant functional descriptions. *Machine Learning*, *1*, 403–422.

Langley, P. (1979). Rediscovering physics with BACON.3. *Proceedings of the Sixth International Joint Conference on Artificial Intelligence* (pp. 505–507). Tokyo, Japan: Morgan Kaufmann.

Langley, P. (1998). The computer-aided discovery of scientific knowledge. *Proceedings of the First International Conference on Discovery Science*. Fukuoka, Japan: Springer.

Langley, P., Simon, H. A., Bradshaw, G. L., & Żytkow, J. M. (1987). *Scientific discovery: Computational explorations of the creative processes*. Cambridge, MA: MIT Press.

Lenat, D. B. (1977). Automated theory formation in mathematics. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence* (pp. 833–842). Cambridge, MA: Morgan Kaufmann.

Ourston, D., & Mooney, R. (1990). Changing the rules: A comprehensive approach to theory refinement. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 815–820). Boston: AAAI Press.

Potter C. S., & Klooster, S. A. (1997). Global model estimates of carbon and nitrogen storage in litter and soil pools: Response to change in vegetation quality and biomass allocation. *Tellus*, *49B*, 1–17.

Potter, C. S., & Klooster, S. A. (1998). Interannual variability in soil trace gas (CO_2, N_2O, NO) fluxes and analysis of controllers on regional to global scales. *Global Biogeochemical Cycles*, *12*, 621–635.

Saito, K., & Nakano, R. (1997). Law discovery using neural networks. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence* (pp. 1078–1083). Yokohama: Morgan Kaufmann.

Saito, K., & Nakano, R. (2000). Discovery of nominally conditioned polynomials using neural networks, vector quantizers and decision trees. *Proceedings of the Third International Conference on Discovery Science* (pp. 325–329). Kyoto: Springer.

Schwabacher, M., & Langley, P. (2001). Discovering communicable scientific knowledge from spatio-temporal data. *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 489–496). Williamstown: Morgan Kaufmann.

Thornthwaite, C. W. (1948) An approach toward rational classification of climate. *Geographic Review*, *38*, 55–94.

Todorovski, L., & Džeroski, S. (1997). Declarative bias in equation discovery. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 376–384). San Francisco: Morgan Kaufmann.

Towell, G. (1991). *Symbolic knowledge and neural networks: Insertion, refinement, and extraction.* Doctoral dissertation, Computer Sciences Department, University of Wisconsin, Madison.

Washio, T. & Motoda, H. (1998). Discovering admissible simultaneous equations of large scale systems. *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (pp. 189–196). Madison, WI: AAAI Press.

Żytkow, J. M., Zhu, J., & Hussam, A. (1990). Automated discovery in a chemistry laboratory. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 889–894). Boston, MA: AAAI Press.

An Efficient Derivation for Elementary Formal Systems Based on Partial Unification*

Noriko Sugimoto, Hiroki Ishizaka, and Takeshi Shinohara

Department of Artificial Intelligence
Kyushu Institute of Technology
Kawazu 680-4, Iizuka 820-8502, Japan
{sugimoto, ishizaka, shino}@ai.kyutech.ac.jp

Abstract. An EFS is a kind of logic programs expressing various formal languages. We propose an efficient derivation for EFS's called an S-derivation, where every possible unifiers are evaluated at one step of the derivation. In the S-derivation, each unifier is partially applied to each goal clause by assigning variables whose values are uniquely determined from the set of all possible unifiers. This contributes to reduce the number of backtracking, and thus the S-derivation works efficiently. In this paper, the S-derivation is shown to be complete for the class of regular EFS's. We implement an EFS interpreter based on the S-derivation in Prolog programming language, and compare the parsing time with that of DCG provided by the Prolog interpreter. As the results of experiments, we verify the efficiency of the S-derivation for accepting context-free languages.

1 Introduction

In the area of machine learning or discovery science, it is an important issue to develop efficient systems dealing with formal languages under a theoretical background. An *elementary formal system* (*EFS*, for short) is a kind of logic programs over the domain of strings [3,11,15]. The EFS's are well-known to be flexible enough to represent not only classes of languages in Chomsky hierarchy [3] but also binary relations over strings [12,13]. It has been shown that the EFS is suitable to discuss learnability in the framework for inductive inference and machine learning of languages [2,3,9,10]. Mukouchi and Arikawa [8] developed a theoretical framework for machine discovery, where refutability of search space is shown to be the most important factor and one of such refutably learnable classes is the class of length-bounded EFS's. Theoretically, EFS's can be used as working systems as Prolog programs because a derivation based on the resolution principle [7] is also defined for EFS's. In EFS's, a derivation procedure is formalized as an acceptor for formal languages [3,15]. Furthermore, the derivation can be used to generating languages [14]. The purpose of this research

* The research reported here is partially supported by the Telecommunication Adbancement Foundation, Japan.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 350–364, 2001.

is to develop an efficient derivation and construct an EFS interpreter based on the derivation.

Since an EFS deals with strings as its domain, unifications for strings should be computed efficiently at each step of the derivation. However, it is known that the unification problem for strings is computationally hard and the unifier is not always uniquely determined even if it is restricted to the maximally general unifier [5,6]. On the other hand, for the first order terms used in Prolog programming language, the unifier is uniquely determined as the most general unifier. Therefore, in an EFS, backtracking occurs for each selection of unifiers as well as clauses. Harada *et al.* [4] introduced restricted EFS's called *variable-separated EFS's*, where there is no variable successively occurring in any term. In the variable-separated EFS, the number of possible unifiers is decreased, and the derivation works efficiently. However, the size of a variable-separated EFS is possibly to be much larger than that of the non-variable-separated EFS equivalent to it. This causes inefficiency in parsing languages. Here, we introduce another approach to develop an efficient EFS interpreter.

When strings have successive occurrence of variables, the number of unifiers becomes large as pointed out by Harada *et al.* [4]. For example, for the strings xyz of variables and $a_1a_2\cdots a_n$ of constant symbols, they have $O(n^2)$ unifiers, because, for each i $(i = 1, 2, \ldots, n-2)$ and j $(j = i+1, i+2, \ldots, n-1)$, all substitutions replacing x with $a_1a_2\cdots a_i$, y with $a_{i+1}a_{i+2}\cdots a_j$, and z with $a_{j+1}a_{j+2}\cdots a_n$ are unifiers of them. In EFS's, since there are many selections for unifiers at each step of a derivation, it has been difficult to construct an efficient interpreter. Thus, we propose a new approach to evaluate all possible unifiers at one step of the derivation. We formalize a *derivation with sets of unifiers* (an *S-derivation*, for short). In the S-derivation, each unifier is partially applied to each goal clause by assigning variables whose values are uniquely determined from the set of all possible unifiers. The S-derivation is a natural extension of the standard derivation for EFS's, because the set of unifiers can be regarded as the unique unifier in EFS's corresponding to the most general unifier in the first order language. We show that the S-derivation is complete for restricted EFS's called *regular EFS's* which define the class of languages equivalent to that of context-free languages.

We implement an S-derivation for regular EFS's in Prolog programming languages, and verify the efficiency of the S-derivation by comparing the running time of the S-derivation with that of definite clause grammars (DCG's) provided by the Prolog interpreter. In our EFS interpreter, each unifier is efficiently computed by using the Aho-Corasick pattern matching algorithm [1]. The Aho-Corasick algorithm finds all occurrences of patterns on the text in linear time with the length of the text. A regular EFS is suitable to the computation of the unification, because each string in the derivation becomes a substring of the initially given text. Therefore, every unifiers used in a derivation can be computed by only once scanning on the given text. As the results of experiments, we show that the S-derivation using the Aho-Corasick algorithm is efficient with respect to the length of a given text and the number of variables in the EFS.

This paper is organized as follows: In Section 2, we give some notations and definitions including derivation and semantics for EFS's. In Section 3 and 4, we introduce S-derivation, and prove completeness of the S-derivation. In Section 5, we outline the EFS interpreter based on the S-derivation, and show experimental results for typical examples of EFS's, where the S-derivation works efficiently. Finally, we summarize the results of this research, and describe some open problems.

2 Preliminaries

In this section, we give some basic definitions and notations according to [3,14, 15].

2.1 Elementary Formal Systems

For a given set A, the set of all finite strings of symbols from A is denoted by A^*. The empty string is denoted by ε. A^+ denotes the set $A^* - \{\varepsilon\}$.

Let Σ, X, and Π be mutually distinct sets. We assume that Σ is a finite set of *constant symbols*, X is a set of *variables*, and Π is a finite set of *predicate symbols*. Each predicate symbol is associated with a non-negative integer called its *arity*.

A *term* is an element of $(\Sigma \cup X)^+$. A term is said to be *regular*, if every variable occurs at most once in the term. An *atomic formula* (*atom*, for short) is of the form $p(\pi_1, \pi_2, \ldots, \pi_n)$, where p is a predicate symbol with arity n and each π_i is a term $(i = 1, 2, \ldots, n)$. A *definite clause* (*clause*, for short) is of the form $A \leftarrow B_1, \ldots, B_n$ $(n \geq 0)$, where $A, B_1, \ldots, B_n$ are atoms. The atom A and the sequence $B_1, \ldots, B_n$ are called the *head* and the *body* of the clause, respectively. A *goal clause* (*goal*, for short) is of the form $\leftarrow B_1, \ldots, B_n$ $(n \geq 0)$ and the goal with $n = 0$ is called the *empty goal*. An *expression* is a term, an atom, a clause, or a goal. An expression E is said to be *ground*, if E has no variable. For an expression E and a variable x, $var(E)$ and $oc(x, E)$ denote the set of all variables occurring in E, and the number of occurrences of x in E, respectively. An *elementary formal system* (*EFS*, for short) is a finite set of clauses.

A *substitution* θ is a (semi-group) homomorphism from $(\Sigma \cup X)^+$ to itself satisfying the following conditions:

1. $a\theta = a$ for each $a \in \Sigma$, and
2. the set $\{x \in X \mid x\theta \neq x\}$, denoted by $D(\theta)$, is finite.

For a substitution θ, if $D(\theta) = \{x_1, x_2, \ldots, x_n\}$ and $x_i\theta = \pi_i$ for every i $(i = 1, 2, \ldots, n)$, then θ is denoted by the set $\{x_1/\pi_1, x_2/\pi_2, \ldots, x_n/\pi_n\}$. For an expression E and a substitution θ, $E\theta$ is defined as the expression by simultaneously replacing each variable x in E with $x\theta$.

Let (E_1, E_2) be a pair of expressions. Then a substitution θ is said to be a *unifier* of E_1 and E_2 if $E_1\theta = E_2\theta$. The set of all unifiers θ of E_1 and E_2 satisfying $D(\theta) \subseteq var(E_1) \cup var(E_2)$ is denoted by $U(E_1, E_2)$. We say that E_1

and E_2 are *unifiable* if the set $U(E_1, E_2)$ is not empty. An expression E_1 is a *variant* of E_2 if there exist two substitutions θ and δ such that $E_1\theta = E_2$ and $E_2\delta = E_1$.

2.2 The Semantics of EFS's

We give two semantics of EFS's by using *provability relations* and *derivations*. First, we introduce the provability semantics. Let Γ and C be an EFS and a clause. Then, the provability relation $\Gamma \vdash C$ is inductively as follows:

1. If $C \in \Gamma$ then $\Gamma \vdash C$.
2. If $\Gamma \vdash C$ then $\Gamma \vdash C\theta$ for any substitution θ.
3. If $\Gamma \vdash A \leftarrow B_1, \ldots, B_m$ and $\Gamma \vdash B_m \leftarrow$ then $\Gamma \vdash A \leftarrow B_1, \ldots, B_{m-1}$.

A clause C is *provable from* Γ if $\Gamma \vdash C$ holds. The *provability semantics* of the EFS Γ, denoted by $PS(\Gamma)$, is defined as the set of all ground atoms A satisfying that $\Gamma \vdash A \leftarrow$. For an EFS Γ and a unary predicate symbol p, the *language defined by* Γ and p is denoted by $L(\Gamma, p)$, and defined as the set of all strings $w \in \Sigma^+$ such that $p(w) \in PS(\Gamma)$.

The second semantics is based on a *derivation* for EFS's. We assume a *computation rule* R to select an atom from every goal. Let Γ be an EFS, G be a goal, and R be a computation rule. A *derivation from* G is a (finite or infinite) sequence of triplets (G_i, C_i, θ_i) $(i = 0, 1, \ldots)$ which satisfies the following conditions:

1. G_i is a goal, θ_i is a substitution, C_i is a variant of a clause in Γ, and $G_0 = G$.
2. $var(C_i) \cap var(C_j) = \emptyset$ for every i and j $(i \neq j)$, and $var(C_i) \cap var(G_i) = \emptyset$ for every i.
3. If $G_i = \leftarrow A_1, \ldots, A_k$, and A_m is the atom selected by R, then C_i is of the form $A \leftarrow B_1, \ldots, B_n$ satisfying that A and A_m are unifiable, $\theta_i \in U(A, A_m)$, and G_{i+1} is of the following form:
$$(\leftarrow A_1, \ldots, A_{m-1}, B_1, \ldots, B_n, A_{m+1}, \ldots, A_k)\theta_i.$$
The atom A_m is called a *selected atom* of G_i, and G_{i+1} is called a *resolvent* of G_i and C_i by θ_i.

A *refutation* is a finite derivation ending with the empty goal. The *procedural semantics* of an EFS Γ, denoted by $RS(\Gamma)$, is defined as the set of all ground atoms A satisfying that there exists a refutation of Γ from the goal $\leftarrow A$.

It has been shown that $PS(\Gamma) = RS(\Gamma)$ for every EFS Γ [15]. This implies that a string $w \in \Sigma^+$ is in the language defined by an EFS Γ and a predicate symbol p if and only if there exists a refutation of Γ from $\leftarrow p(w)$. Thus, the derivation procedure can be regarded as an acceptor for the language.

Finally, we give the distinct set from an EFS language. Let Γ be an EFS, and (G_i, C_i, θ_i) $(i = 0, 1, \ldots, n)$ be a finite derivation of Γ. The derivation is said to be *finitely failed with the length* n if there exists no clause in Γ such that its head and the selected atom of G_n are unifiable. Furthermore, we define $FFS(\Gamma)$ as the set of all ground atoms A satisfying that all derivations of Γ from $\leftarrow A$ are finitely failed within the length n.

3 Extended Derivations with Sets of Unifiers

In this section, we introduce a *derivation with sets of unifiers* (*S-derivation*, for short). In the S-derivation, each unifier is partially applied to each goal clause by assigning variables whose values are uniquely determined from the set of all possible unifiers. Since there are infinitely many unifiers for terms containing variables, it is difficult to compute the derivation from the goal containing variables. However, for restricted terms, all unifiers are computable by using *maximally general unifiers*. The S-derivation works efficiently by using the maximally general unifiers. Furthermore, in this section, the S-derivation is shown to be complete for accepting and generating languages defined by restricted EFS's called *regular EFS's*.

3.1 Maximally General Unifiers

Let $\theta = \{x_1/\pi_1, x_2/\pi_2, \ldots, x_m/\pi_m\}$ and $\delta = \{y_1/\tau_1, y_2/\tau_2, \ldots, y_n/\tau_n\}$ be substitutions. Then, we define a *composition of* θ and δ as follows:

$$\theta \cdot \delta = \{x_i/\pi_i\delta \mid x_i \neq \pi_i\delta\} \cup \{y_i/\tau_i \mid y_i \notin D(\theta)\}.$$

Let θ, δ and γ be substitutions, and E be an expression. Then, we can prove the following equations along the same line of argument as definite programs [7]:

1. $(E\theta)\delta = E(\theta \cdot \delta)$, and
2. $(\theta \cdot \delta) \cdot \gamma = \theta \cdot (\delta \cdot \gamma)$.

Let V be a finite set of variables, and (θ, δ) be a pair of substitutions. Then, we say that θ and δ are *equivalent on* V, if $\pi\theta$ is a variant of $\pi\delta$ for any $\pi \in (\Sigma \cup V)^+$. We show that the problem of determining whether or not θ and δ are equivalent on V is solvable by the following lemma.

Lemma 1. *Let θ and δ be substitutions, and $V = \{x_1, x_2, \ldots, x_n\}$ be a finite set of variables. Then, θ and δ are equivalent on V if and only if the following statements hold:*

1. *$x\theta$ is a variant of $x\delta$, for every $x \in V$, and*
2. *$x_1x_2 \cdots x_n\theta$ is a variant of $x_1x_2 \cdots x_n\delta$.*

Proof. We can prove this lemma by the induction on the length of $\pi \in (\Sigma \cup V)^+$. □

Let (E_1, E_2) be a pair of expressions. A *maximally general unifier* (*mxgu*, for short) of E_1 and E_2 is a unifier $\theta \in U(E_1, E_2)$ satisfying that, for any $\delta \in U(E_1, E_2)$ such that θ and δ are equivalent on $var(E_1) \cup var(E_2)$, there is no substitution γ such that $\theta = \delta \cdot \gamma$. The set of all mxgu's of E_1 and E_2 is denoted by $MXGU(E_1, E_2)$.

For two terms π and τ, we define the number of mxgu's of π and τ as the cardinality of equivalence classes of substitutions on $var(\pi) \cup var(\tau)$. Thus, we say that $MXGU(\pi, \tau)$ *is finite*, if the number of mxgu's is finite without equivalent substitutions on $var(\pi) \cup var(\tau)$. From the definition of maximally general unifiers, the following lemmas hold [5,6,14].

Lemma 2. *Let π and τ be regular terms such that $var(\pi) \cap var(\tau) = \emptyset$. Then, the set $MXGU(\pi, \tau)$ is finite and computable.*

Lemma 3. *Let π and τ be terms. If π is ground, then the set $MXGU(\pi, \tau)$ is finite and computable, and $MXGU(\pi, \tau) = U(\pi, \tau)$ holds.*

Lemma 4. *Let x be a variable and π be a term which does not include x. Then, $MXGU(\pi, x)$ is a singleton set which consists of the substitution $\{x/\pi\}$.*

3.2 S-Derivation

In the following argument, we assume that every substitution θ satisfies $\theta \cdot \theta = \theta$, that is, $var(x\theta) \cap D(\theta) = \emptyset$ for every variable $x \in D(\theta)$.

Definition 1. *For two substitutions θ and δ, we define $\theta \circ \delta$ as the set of all substitutions σ satisfying that $\sigma = \theta \cdot \delta \cdot \gamma = \delta \cdot \theta \cdot \gamma$ for some substitution γ.*

Note that, for each element σ of the set $\theta \circ \delta$, $x\sigma$ becomes the element of the intersection of sets of strings which are unifiable with $x\theta$ and $x\delta$.

Substitutions θ and δ are said to be *inconsistent* if $\theta \circ \delta = \emptyset$, and *consistent*, otherwise. We define $MIN(\theta \circ \delta)$ as the minimum subset of $\theta \circ \delta$ satisfying that, for any $\sigma \in \theta \circ \delta$, there exists $\sigma' \in MIN(\theta \circ \delta)$ such that $\sigma = \sigma' \cdot \gamma$ for some substitution γ.

For two finite sets Θ and Δ of substitutions, we define

1. $MIN(\Theta \circ \Delta) = \bigcup_{(\theta,\delta) \in \Theta \times \Delta} MIN(\theta \circ \delta)$, and
2. $INT(\Theta) = \bigcap_{\theta \in \Theta} \theta$.

Lemma 5. *Let θ and δ be substitutions. If δ is ground, then the set $MIN(\theta \circ \delta)$ is finite and computable.*

Proof. Let θ and δ be substitutions $\{x_i/\pi_i \mid i \in \{1, 2, \ldots, m\}\}$ and $\{y_i/t_i \mid i \in \{1, 2, \ldots, n\}\}$, respectively.

If $\sigma \in \theta \circ \delta$ then there exists a substitution γ satisfying that

1. $\sigma = \theta \cdot \delta \cdot \gamma = \{x_i/\pi_i\delta\gamma \mid i = 1, 2, \ldots, m\} \cup \delta \cup \gamma$, and
2. $\pi_i\delta\gamma = t_j$ for every $x_i = y_j \in D(\theta) \cap D(\delta)$,

from Definition 1. Let S be the set of all possible γ satisfying the above conditions and $D(\gamma) \subseteq var(\pi_1\delta) \cup var(\pi_2\delta) \cup \cdots \cup var(\pi_m\delta)$. Since, from Lemma 3, the set $U(\pi_i\delta, t_j)$ is finite for each $x_i = y_j \in D(\theta) \cap D(\delta)$, the set S is also finite and computable. It is clear that $\sigma = \theta \cdot \delta \cdot \gamma \in MIN(\theta \circ \delta)$ for each $\gamma \in S$, because every $\gamma \in S$ is ground. Futhermore, we can show that, for every substitution γ', if $\theta \cdot \delta \cdot \gamma' = \delta \cdot \theta \cdot \gamma'$ holds, then there exists $\gamma \in S$ such that $\gamma' = \gamma \cdot \gamma''$. Thus, the set $MIN(\theta \circ \delta)$ consists of $\theta \cdot \delta \cdot \gamma$ for every $\gamma \in S$. It is clear that $MIN(\theta \circ \delta)$ is finite and computable. □

Example 1. We consider the substitutions: $\theta_1 = \{x/y\}$, $\theta_2 = \{x/aaz, y/az\}$, $\theta_3 = \{x/aaz, y/zb\}$, and $\delta = \{x/aaa, y/ab\}$.

From $x\theta_1\delta = y\delta = ab$ and $x\delta = aaa$, the set $U(x\theta_1\delta, x\delta)$ is empty. Thus, $\theta_1 \circ \delta = \emptyset$, and θ and δ are inconsistent.

From $x\theta_2\delta = aaz\delta = aaz$ and $x\delta = aaa$, $U(x\theta_2\delta, x\delta)$ is the set $\{\{z/a\}\}$. From $y\theta_2\delta = az\delta = az$ and $y\delta = ab$, $U(y\theta_2\delta, y\delta)$ is the set $\{\{z/b\}\}$. Then, there exists no substitution γ such that $x\theta_2\delta\gamma = aaa$ and $y\theta_2\delta\gamma = ab$. Therefore, $\theta_2 \circ \delta = \emptyset$, and θ_2 and δ are inconsistent.

From $x\theta_3\delta = aaz\delta = aaz$ and $x\delta = aaa$, $U(x\theta_3\delta, x\delta)$ is the set $\{\{z/a\}\}$. From $y\theta_3\delta = zb\delta = zb$ and $y\delta = ab$, $U(y\theta_3\delta, y\delta)$ is the set $\{\{z/a\}\}$. Then, only the substitution $\gamma = \{z/a\}$ satisfies $x\theta_3\delta\gamma = aaa$ and $y\theta_3\delta\gamma = ab$. Therefore, $MIN(\theta \circ \delta)$ has only one element $\theta_3 \cdot \delta \cdot \gamma = \{x/aaa, y/ab, z/a\}$, and θ_3 and δ are consistent.

Lemma 6. *Let $\theta = \{x_i/\pi_i \mid i = 1, 2, \ldots, m\}$ and $\delta = \{y/\tau\}$ be substitutions satisfying that, for each i $(i = 1, 2, \ldots, m)$, π_i and τ are regular, and $var(\pi_i) \cap var(\tau) = D(\theta) \cap var(\tau) = \emptyset$. Then, the set $MIN(\theta \circ \delta)$ is finite and computable.*

Proof. If $y \notin D(\theta)$ then $\delta \cdot \theta \cdot \delta = \theta \cdot \delta$ and $\theta \cdot \delta \cdot \delta = \theta \cdot \delta$ from the assumption and $\delta \cdot \delta = \delta$. Thus, $\theta \cdot \delta \in \theta \circ \delta$ holds. Furthermore, for every $\sigma \in \theta \circ \delta$, $\sigma = \theta \cdot \delta \cdot \gamma$ holds for some substitution γ. Thus, the set $MIN(\theta \circ \delta)$ is a singleton set $\{\theta \cdot \delta\}$.

If $y = x_k \in D(\theta)$ then $y \notin var(\pi_i)$ for every i $(i = 1, 2, \ldots, m)$ from the assumption $\theta \cdot \theta = \theta$. Thus, $\theta \cdot \delta = \theta$ holds. Furthermore, from the assumption of this lemma, $var(\tau) \cap D(\theta) = \emptyset$ and $\delta \cdot \theta = \delta \cup \{x_i/\pi_i \mid i \neq k\}$. If π_k and τ are not unifiable, then θ and δ are inconsistent. Otherwise, for any $\gamma \in MXGU(\pi_k, \tau)$, $\theta \cdot \gamma \in \theta \circ \delta$, because $\theta \cdot \delta \cdot \gamma = \delta \cdot \theta \cdot \gamma = \theta \cdot \gamma$ holds. Furthermore, from the definition of mxgu's, for any substitution σ such that $\theta \cdot \delta \cdot \sigma \in \theta \circ \delta$, there exists $\gamma \in MXGU(\pi_k, \tau)$ satisfying that $\sigma = \gamma \cdot \gamma'$ for some substitution γ'. Thus, $MIN(\theta \circ \delta)$ is the set $\{\theta \cdot \gamma \mid \gamma \in MXGU(\pi_k, \tau)\}$. Since the set $MXGU(\pi_k, \tau)$ is finite and computable from Lemma 2, $MIN(\theta \circ \delta)$ is also finite and computable. □

Example 2. Let $\theta_1 = \{x/aya, z/y\}$, $\theta_2 = \{y/aza\}$, and $\delta = \{y/y_1y_2\}$. Then, $MIN(\theta_1 \circ \delta)$ is a singleton set which consists of $\theta \cdot \delta = \{x/ay_1y_2a, y/y_1y_2, z/y_1y_2\}$. On the other hand, since

$$MXGU(aza, y_1y_2) = \left\{ \begin{array}{l} \{y_1/a, y_2/za\} \\ \{y_1/az, y_2/a\} \\ \{y_1/az_1, y_2/z_2a, z/z_1z_2\} \end{array} \right\},$$

we can obtain the following set:

$$MIN(\theta_2 \circ \delta) = \left\{ \begin{array}{l} \{y/aza, y_1/a, y_2/za\} \\ \{y/aza, y_1/az, y_2/a\} \\ \{y/az_1z_2a, y_1/az_1, y_2/z_2a, z/z_1z_2\} \end{array} \right\}.$$

Definition 2. Let Γ be an EFS, G be a goal of Γ, and R be a computation rule. An *S-derivation* from G is a (finite or infinite) sequence of triplets (G_i, C_i, Θ_i) $(i = 0, 1, \ldots)$ which satisfies the following conditions:

1. G_i is a goal, Θ_i is a finite set of substitutions, C_i is a variant of a clause in Γ, and $G_0 = G$.
2. $var(C_i) \cap var(C_j) = \emptyset$ for every i and j $(i \neq j)$, and $var(C_i) \cap var(G_i) = \emptyset$ for every i.
3. Let $G_i = \leftarrow A_1, \ldots, A_k$, $C_i = A \leftarrow B_1, \ldots, B_q$, and A_m is the selected atom of G_i. If $i = 0$, then $\Theta_i = MXGU(A_m, A)$. Otherwise, $\Theta_i = MIN(\Theta_{i-1} \circ MXGU(A_m, A))$ for each i. The next goal G_{i+1} is of the following form:
$$(\leftarrow A_1, \ldots, A_{m-1}, B_1, \ldots, B_q, A_{m+1}, \ldots, A_k)INT(\Theta_i).$$

If the S-derivation ends with the empty goal G_n, then it is said to be an *S-refutation from* G, and each substitution in Θ_{n-1} is called an *answer substitution for* G by Γ.

Definition 3. Let Γ be an EFS, and (G_i, C_i, Θ_i) $(i = 0, 1, \ldots, n)$ be a finite S-derivation of Γ. The derivation is said to be *finitely failed with the length* n if

1. $\Theta_n = \emptyset$, or
2. there exists no clause in Γ such that its head and the selected atom of G_n are unifiable.

For an EFS Γ, we define the following two sets: $SFFS(\Gamma)$ is the set of all ground atoms A satisfying that all S-derivations of Γ from $\leftarrow A$ are finitely failed within the length n, and $SRS(\Gamma)$ is the set of all ground atoms A satisfying that there exists an S-refutation of Γ from $\leftarrow A$.

3.3 Completeness of S-Derivation

An EFS Γ is said to be *regular* if all predicate symbols in Γ are unary, and each clause $A \leftarrow B_1, B_2, \ldots, B_n$ in Γ satisfying the following conditions:

1. the term in A is regular,
2. every term in $B_1, B_2, \ldots, B_n$ are mutually distinct variables, and
3. $var(B_1) \cup var(B_2) \cup \cdots \cup var(B_n) \subseteq var(A)$.

It has been shown that the class of languages defined by regular EFS's is equivalent to that of context-free languages [3]. For the regular EFS, we show that an S-derivation is complete by the following theorem.

Theorem 1. *For every regular EFS* Γ, $PS(\Gamma) = RS(\Gamma) = SRS(\Gamma)$ *holds.*

The above theorem can be proved by the following lemmas and proposition.

Lemma 7. *Let* Γ *be a regular EFS,* G_0 *be a ground goal, and* (G_i, C_i, Θ_i) $(i = 0, 1, \ldots, n)$ *be an S-derivation from* G_0. *Then, for every* $\sigma \in \Theta_{n-1}$, σ *is ground, and there exists a derivation* (G'_i, C_i, θ_i) $(i = 0, 1, \ldots, n)$ *such that* $G'_0 = G_0$, *and* $G'_i = G_i\sigma$ *for each* i $(i = 1, 2, \ldots, n)$.

Proof. Let $p(w)$ be an selected atom of G_0, and $p(\pi)$ be the head of C_0. Then, from the definition of an S-derivation, $\Theta_0 = U(w, \pi)$ holds. Furthermore, for every $\sigma \in \Theta_0$, $INT(\Theta_0) \subseteq \sigma$ holds. Let G'_1 be the resolvent of G_0 and C_0 by σ. Then, the derivation $(G_0, C_0, \sigma), (G'_1, \_, \_)$ satisfies the statement.

Next, we assume that $p(\tau)$ be an selected atom of G_{n-1}, and $p(\pi)$ be the head of C_{n-1}. Then, from the definitions of an S-derivation and a regular EFS, τ is a gound term $w \in \Sigma^+$ or a variable $x \in D(\sigma_{n-2})$ for every $\sigma_{n-2} \in \Theta_{n-2}$. If $\sigma \in \Theta_{n-1}$ then there exists $\sigma_{n-2} \in \Theta_{n-2}$ such that $\sigma \in \sigma_{n-2} \circ \delta$ for some $\delta \in MXGU(\tau, \pi)$.

If the selected atom is $p(w)$ then δ is ground. Thus, σ is also ground. If the selected atom is $p(x)$ then $\delta = \{x/\pi\}$ from Lemma 4. Since $x/w \in \sigma_{n-2}$ for some $w \in \Sigma^+$, $\sigma \in \sigma_{n-2} \circ \delta = \{\sigma_{n-2} \cdot \gamma \mid \gamma \in MXGU(w, \pi)\}$. Thus, σ is ground.

From the assumption of the induction, there exists a derivation (G'_i, C_i, θ_i) $(i = 0, 1, \ldots, n-1)$ such that $G'_0 = G_0$, and $G'_i = G_i\sigma_{n-2}$ for each i $(i = 1, 2, \ldots, n-1)$. Since σ_{n-2} is ground, it is clear that $\sigma_{n-2} \subseteq \sigma$. Let G'_n be the resolvent of G'_{n-1} and C_{n-1} by $\theta_{n-1} \in U(w, \pi)$, then it is clear that the derivation (G'_i, C_i, θ_i) $(i = 0, 1, \ldots, n)$ satisfies the statement. □

Lemma 8. *Let Γ be a regular EFS, G_0 be a ground goal, and (G_i, C_i, θ_i) $(i = 0, 1, \ldots)$ be a derivation from G_0. Then, there exists an S-derivation (G'_i, C_i, Θ_i) $(i = 0, 1, \ldots)$ and a substitution $\sigma_i \in \Theta_i$ such that $G'_0 = G_0$, and $G'_{i+1}\sigma_i = G_{i+1}$ for each i $(i = 1, 2, \ldots)$.*

Proof. Let $p(w)$ be an selected atom of G_0, and $p(\pi)$ be the head of C_0. Then, from the definition of a derivation, $\theta_0 \in U(w, \pi)$. On the other hand, from the definition of an S-derivation, $\Theta_0 = MXGU(w, \pi) = U(w, \pi)$. Thus, $\theta_0 \in \Theta_0$, and $G'_1\theta_0 = G_1$.

Next, we assume that there exists $\sigma_{k-1} \in \Theta_{k-1}$ such that $G'_k\sigma_{k-1} = G_k$. Let $p(w)$ be an selected atom of G_k, and the head of C_k be $p(\pi)$. Then, from the definition of a derivation, $\theta_k \in U(w, \pi)$. On the other hand, from the definition of an S-derivation and the assumption of the induction, the selected atom of G'_k has the form $p(w)$ or $p(x)$ for some $w \in \Sigma^+$ and $x \in D(\sigma_{k-1})$.

If the selected atom is $p(w)$ then $\Theta_k = MIN(\Theta_{k-1} \circ U(w, \pi))$. Since $\sigma_{k-1} \in \Theta_{k-1}$ and $\theta_k \in U(w, \pi)$, $\sigma_{k-1} \circ \theta_k \subseteq \Theta_k$ holds. Furthermore, from the definitions of an S-derivation, $D(\sigma_{k-1}) \cap D(\theta_k) = \emptyset$. Thus, σ_{k-1} and θ_k are consistent, and $\sigma_{k-1} \cup \theta_k \in \sigma_{k-1} \circ \theta_k$.

If the selected atom is $p(x)$ then $\Theta_k = MIN(\Theta_{k-1} \circ MXGU(x, \pi))$, and $MXGU(x, \pi)$ is a singleton set $\{\{x/\pi\}\}$ from Lemma 4. Since $\sigma_{k-1} \in \Theta_{k-1}$, $MIN(\sigma_{k-1} \circ \{x/\pi\}) \subseteq \Theta_k$ holds. Furthermore, from $x/w \in \sigma_{k-1}$ and $U(w, \pi) \neq \emptyset$, $\{x/\pi\}$ and σ_{k-1} are consistent. From the statement in the proof of Lemma 6, $MIN(\{x/\pi\} \circ \sigma_{k-1}) = \{\sigma_{k-1} \cdot \gamma \mid \gamma \in MXGU(w, \pi)\}$. Since $\theta_k \in MXGU(w, \pi)$, $\sigma_{k-1} \cdot \theta_k = \sigma_{k-1} \cup \theta_k \in \Theta_k$.

It is clear that $\sigma_{k-1} \cup \theta_k$ becomes a substitution, and satisfies the statement. □

From Lemma 7 and 8, we can prove the following proposition.

Proposition 1. *Let Γ be a regular EFS, G_0 be a ground goal. Then, there exists a refutation from G_0 if and only if there exists an S-refutation from G_0.*

Furthermore, we can also prove the following theorem.

Theorem 2. *For every regular EFS Γ, $FFS(\Gamma) = SFFS(\Gamma)$ holds.*

Example 3. For an EFS

$$\Gamma = \left\{ \begin{array}{l} (1)\ p(xy) \leftarrow q(x), r(y); \\ (2)\ q(a^n) \leftarrow; \\ (3)\ r(aa) \leftarrow \end{array} \right\},$$

and a goal $\leftarrow p(a^{n+1})$, Fig 1 describes the derivation and the S-derivation as trees like SLD-trees [7]. In the derivation and the S-derivation in Fig 1, the label (k, θ) on each edge represents the derivation by the clause (k) and the unifier or the set of unifiers θ. The derivation needs $n+1$ times backtracking to determine $p(a^{n+1}) \in FF(\Gamma)$. On the other hand, in the S-derivation, it is determined by twice backtracking.

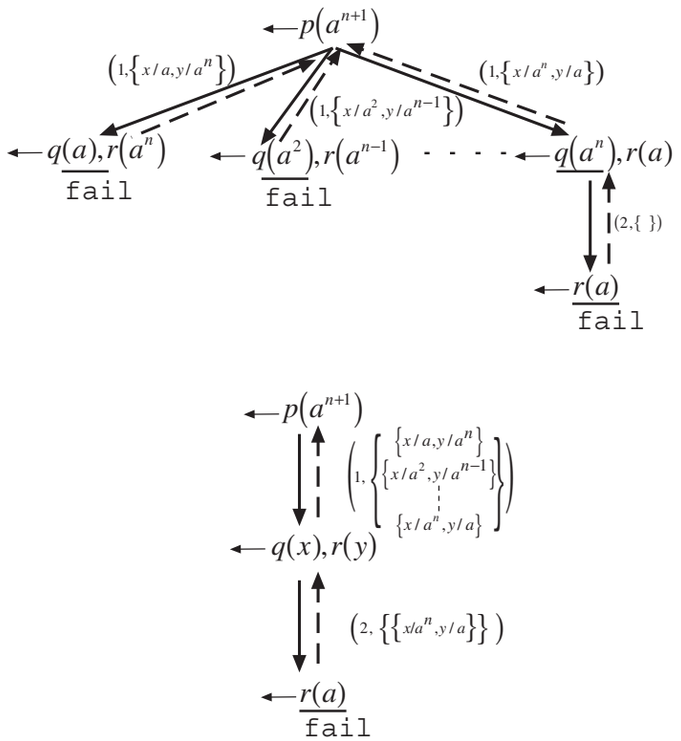


Fig. 1. Backtracking by a derivation and an S-derivation

4 An Implementation of EFS Interpreter

In this section, we outline an implementation of EFS interpreter based on an S-derivation, and give some results of experiments for typical examples of EFS's where the S-derivation works efficiently.

In order to construct an efficient interpreter, we adopt two ideas:

1. computing each unifiers by using the Aho-Corasick pattern matching algorithm, and
2. reducing the number of backtracking of a derivation by using an S-derivation.

4.1 Unifications by the Aho-Corasick Pattern Matching Algorithm

The Aho-Corasick pattern matching algorithm finds all occurrence positions of some patterns by scanning the given text. From a given EFS, the EFS interpreter makes a pattern matching machine in advance for all ground strings in the EFS. For each given ground goal, the pattern matching machine scans the ground term in the given goal clause and outputs all occurrence positions of patterns on the term. From the occurrence positions, each unifier is efficiently computed.

Example 4. Let $w = aaabaaabaaabaaa$ and $\tau = xbyabz$ be terms, where $a, b \in \Sigma$ and $x, y, z \in X$. For constant substrings b and ab of τ, the pattern matching machine finds the occurrence positions on w as follows:

$$\begin{aligned} b &: (4:4), (8:8), (12:12), \\ ab &: (3:4), (7:8), (11:12), \end{aligned}$$

where, $(i:j)$ in the line of b (*resp.* ab) means that the substring from ith to jth of w is b (*resp.* ab). For each occurrence $(i_b : j_b)$ of b and $(i_{ab} : j_{ab})$ of ab such that $j_b \leq i_{ab}$, we obtain unifiers of w and τ as follows:

$$\begin{aligned} &\{x/(1:3), y/(5:6), z/(9:15)\} && \text{from } ((4:4), (7:8)), \\ &\{x/(1:3), y/(5:10), z/(13:15)\} && \text{from } ((4:4), (11:12)), \\ &\{x/(1:7), y/(8:10), z/(13:15)\} && \text{from } ((8:8), (11:12)). \end{aligned}$$

A regular EFS is suitable to this computation of the unifier, because every ground term in each resolvent of the derivation becomes a substring of the term in the given initial goal. This implies that all unifiers used in the derivation can be computed by only once scanning the given initial goal by the pattern matching machine.

4.2 An Implementation of S-Derivation

Since an S-derivation deals with the set of all possible unifiers at each step of the derivation, it is important to adopt a compact representation of the set. From the property of a regular EFS, all terms in the derivation are substrings of the term in the given initial goal. Thus, the set of all possible unifiers can be divided into some parts as shown by the next example.

Example 5. Let $w = aabaabaabaa$ and $\tau = xybz$ be terms, where $a, b \in \Sigma$ and $x, y, z \in X$. Then, the set of all unifiers

$$U(w,\tau) = \left\{ \begin{array}{l} \{x/a, y/a, z/aabaabaa\}, \{x/a, y/abaa, z/aabaa\}, \\ \{x/aa, y/baa, z/aabaa\}, \{x/aab, y/aa, z/aabaa\}, \\ \{x/aaba, y/a, z/aabaa\}, \{x/a, y/abaabaa, z/aa\}, \\ \{x/aa, y/baabaa, z/aa\}, \{x/aab, y/aabaa, z/aa\}, \\ \{x/aaba, y/abaa, z/aa\}, \{x/aabaa, y/baa, z/aa\}, \\ \{x/aabaab, y/aa, z/aa\}, \{x/aabaaba, y/a, z/aa\} \end{array} \right\}$$

can be divided into these three parts:

$$U_1 = \left\{ \{x/a, y/a, z/aabaabaa\} \right\},$$

$$U_2 = \left\{ \begin{array}{l} \{x/a, y/abaa, z/aabaa\}, \\ \{x/aa, y/baa, z/aabaa\}, \\ \{x/aab, y/aa, z/aabaa\}, \\ \{x/aaba, y/a, z/aabaa\} \end{array} \right\}, \quad U_3 = \left\{ \begin{array}{l} \{x/a, y/abaabaa, z/aa\}, \\ \{x/aa, y/baabaa, z/aa\}, \\ \{x/aab, y/aabaa, z/aa\}, \\ \{x/aaba, y/abaa, z/aa\}, \\ \{x/aabaa, y/baa, z/aa\}, \\ \{x/aabaab, y/aa, z/aa\}, \\ \{x/aabaaba, y/a, z/aa\} \end{array} \right\}.$$

Furthermore, each U_i $(i = 1, 2, 3)$ is represented as follows:

$$U_1 = \{\{x/(1:1), y/(2:2), z/(4:11)\}\},$$
$$U_2 = \{\{x/(1:k), y/(k+1:5), z/(7:11)\} \mid 4 \geq k \geq 1\},$$
$$U_3 = \{\{x/(1:k), y/(k+1:8), z/(10:11)\} \mid 7 \geq k \geq 1\},$$

where each $(i:j)$ represents the substring from ith to jth of w. For an EFS $\Gamma = \{p(xybz) \leftarrow q_1(x), q_2(y), q_3(z);\ q_1(aa) \leftarrow;\ q_2(baa) \leftarrow;\ q_3(aabaa) \leftarrow\}$ and the set U_2, the S-derivation from $\leftarrow p(w)$ is shown in Fig 2.

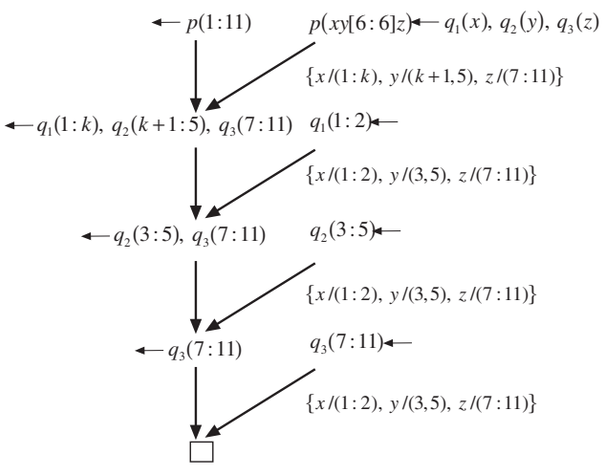


Fig. 2. An S-derivation from the goal $\leftarrow p(w)$.

We can easily show that the derivation with the divided sets on unifiers is equivalent to the S-derivation. From the occurrence positions given by the pattern matching machine, each set U_i is efficiently computed. Thus, an S-derivation is efficient.

4.3 Experimental Results

We construct three types of EFS interpreters C_1, C_2, and C_3, where C_1, C_2, and C_3 use a derivation with naive unifications, a derivation with unifications by the Aho-Corasick algorithm, and an S-derivation with the unification by the Aho-Corasick algorithm, respectively. We verify the efficiency of the S-derivation and unifications using the Aho-Corasick algorithm, by comparing the running time of these interpreter with that of the definite clause grammar (DCG) provided by the Prolog interpreter.

We consider the following EFS's and DCG's:

$$\Gamma_1 = \left\{ \begin{array}{l} p_0(x_1x_2) \leftarrow p_1(x_1), p_2(x_2); \\ p_1(aaxaa) \leftarrow p_1(x); \\ p_1(bbxbb) \leftarrow p_1(x); \\ p_1(aaaa) \leftarrow; \quad p_1(bbbb) \leftarrow; \\ p_2(a) \leftarrow; \quad p_2(aa) \leftarrow . \end{array} \right\}, D_1 = \left\{ \begin{array}{l} p_0 \rightarrow p_1, p_2; \\ p_1 \leftarrow aap_1aa; \\ p_1 \leftarrow bbp_1bb; \\ p_1 \rightarrow aaaa; \quad p_1 \rightarrow bbbb; \\ p_2 \rightarrow a; \quad p_2 \rightarrow aa. \end{array} \right\},$$

$$\Gamma_2 = \left\{ \begin{array}{l} p_0(x_1x_2x_3x_4aaa) \leftarrow \\ \quad p_1(x_1), p_1(x_2), p_1(x_3), p_1(x_4); \\ p_0(aaa) \leftarrow; \\ p_1(axa) \leftarrow p_1(x); \\ p_1(bxb) \leftarrow p_1(x); \\ p_1(a) \leftarrow; \quad p_1(b) \leftarrow; \\ p_1(aa) \leftarrow; \quad p_1(bb) \leftarrow . \end{array} \right\}, D_2 = \left\{ \begin{array}{l} p_0 \rightarrow p_1p_1p_1p_1aaa; \\ p_0 \rightarrow aaa; \\ p_1 \rightarrow ap_1a; \\ p_1 \rightarrow bp_1b; \\ p_1 \rightarrow a; \quad p_1 \rightarrow b; \\ p_1 \rightarrow aa; \quad p_1 \rightarrow bb. \end{array} \right\}.$$

The DCG D_i and the EFS Γ_i represent the same language ($i = 1, 2$).

Table 1. The running time for the EFS Γ_1 and the DCG D_1 (sec.)

The length of the text	C_1	C_2	C_3	DCG
100	18.17	50.46	2.03	0.2
200	64.05	195.12	3.93	0.4
300	137.86	435.54	5.86	0.54
400	238.4	762.86	7.78	0.76
500	367.11	1181.39	9.62	0.89

The running time by EFS interpreters for Γ_1 and the DCG for D_1 are shown in Table 1. The input data consist of 30 strings from $\{a, b\}$. From the results of this experiment, If an EFS has successive occurrence of variables, then an S-derivation is more efficient than the derivation as shown by the difference between the running time of C_2 and C_3.

Table 2. The running time for the EFS Γ_2 and the DCG D_2 (sec.)

The length of the text	C_1	C_2	C_3	DCG
5	8.71	8.75	9.25	6.49
10	88.42	17.42	20.05	22.43
15	473.15	52.62	73.24	115.5
20	1619.83	168.24	351.96	528.6
25	4200.69	424.1	1175.22	1648.99

In Table 2, we present the running time of each EFS interpreter and DCG, for Γ_2 and D_2. The input data consist of 1000 strings from $\{a, b\}$. The unification by the Aho-Corasick algorithm is efficient as the difference between the running time of C_1 and C_2. Furthermore, we find C_2 and C_3 are more efficient than DCG. This result represens that the number of backtracking by the EFS interpreter are less than that by the DCG.

5 Conclusion

We have proposed an efficient derivation for EFS's called S-derivation, where all possible unifiers are evaluated at one step of the derivation. We have shown that the S-derivation is complete for accepting context-free languages. Furthermore, we have implemented the S-derivation, and verified its efficiency by comparing with the running time of DCG's.

One of the open problems is to discuss computability of the S-derivation for the extended classes of regular EFS's. Since, in the S-derivation, each resolvent contains variables even if the initial goal is ground, the unification should be efficiently computed for terms containing variables. However, it is known that the unification problem for non-regular terms is NP-complete. Therefore, we have to consider another approach for the extended class of EFS's.

The S-derivation can be applied to translations over strings. We have already constructed the translator for regular TEFS's [12] which represent binary relations over context-free languages. It is a future work to formalize generating languages by the S-derivation in the framework of TEFS's, and to design a translator for real data by using our results.

References

1. A. V. Aho and M. J. Corasick: *Efficient string matching : An aid to bibliographic search*, Communication of the ACM 18, No.6, 333–340 (1975).
2. S. Arikawa, S. Miyano, A. Shinohara, T. Shinohara, and A. Yamamoto: *Algorithmic learning theory with elementary formal systems*, IEICE Transaction on Information and Systems E75-D, 405–414 (1992).
3. S. Arikawa, T. Shinohara, and A. Yamamoto: *Learning elementary formal systems*, Theoretical Computer Science 95, 97–113 (1992).

4. N. Harada, S. Arikawa, and H. Ishizaka: *A Class of elementary formal systems that has an efficient parsing algorithm*, Information Modeling and Knowledge Bases IX, 89–101 (1997).
5. J. Jaffar: *Minimal and complete word unification*, Journal of the ACM 37, 47–85 (1990).
6. D. Kapur: *Complexity of unification problems with associative-commutative operation*, Journal of Automated Reasoning 9, 261–288 (1992).
7. J. W. Lloyd: *Foundations of logic programming (second edition)*, Springer-Verlag (1987).
8. Y. Mukouchi and S. Arikawa: *Towards a mathematical theory of machine discovery from facts*, Theoretical Computer Science 137, 53–84 (1995).
9. T. Shinohara: *Inductive inference on monotonic formal systems from positive data*, New Generation Computing 8, 371–384 (1991).
10. T. Shinohara: *Rich classes inferable from positive data: Length-bounded elementary formal system*, Information and Computation 108, 175–186 (1994).
11. R. Smullyan: *Theory of formal systems*, Princeton Univ. Press, Princeton (1961).
12. N. Sugimoto, K. Hirata and H. Ishizaka: *Constructive learning of translations based on dictionaries*, In Proceedings of the Seventh International Workshop on Algorithmic Learning Theory, Lecture Notes in Artificial Intelligence 1160, 177–184 (1996).
13. N. Sugimoto: *Learnability of translations from positive examples*, In Proceedings of the Ninth International Conference on Algorithmic Learning Theory, Lecture Notes in Artificial Intelligence 1501, 169–178 (1998).
14. N. Sugimoto and H. Ishizaka: *Generating languages by a derivation procedure for elementary formal systems*, Information Processing Letters 69, 161–166 (1999).
15. A. Yamamoto: *Procedural semantics and negative information of elementary formal system*, Journal of Logic Programming 13, 89–97 (1992).

Worst-Case Analysis of Rule Discovery

Einoshin Suzuki

Electrical and Computer Engineering, Yokohama National University,
79-5 Tokiwadai, Hodogaya, Yokohama 240-8501, Japan
suzuki@dnj.ynu.ac.jp

Abstract. In this paper, we perform a worst-case analysis of rule discovery. A rule is defined as a probabilistic constraint of true assignment to the class attribute of corresponding examples. In data mining, a rule can be considered as representing an important class of discovered patterns. We accomplish the aforementioned objective by extending a preliminary version of PAC learning, which represents a worst-case analysis for classification. Our analysis consists of two cases: the case in which we try to avoid finding a bad rule, and the case in which we try to avoid overlooking a good rule. Discussions on related works are also provided for PAC learning, multiple comparison, analysis of association rule discovery, and simultaneous reliability evaluation of a discovered rule.

1 Introduction

Data mining [2] can be defined as extraction of useful knowledge from massive data, and is gaining increasing attention due to advancement of various information technologies. Data mining can be regarded as advanced data analysis, and a typical process of analysis consists of several steps [2]. Pattern extraction represents an important step in such a process. A rule is defined as a probabilistic constraint inherent in a data set, and is widely recognized as representing one of the most important patterns in data mining.

Although rule discovery has been extensively studied in data mining, its theoretical analyses are surprisingly rare. Several exceptions include Agrawal et al.'s analysis of association rule discovery [1] and our analysis of a discovered rule based on simultaneous reliability evaluation [10]. However, these studies ignore the total number of rules that can be discovered from a data set. This fact represents that these studies fail to relate the size of a discovery problem to the number of examples needed for successful discovery, and suggests that a more solid foundation of data mining should be established.

As a first step toward this objective, we extend a preliminary version of PAC learning [7], which represents a worst-case analysis of classification. Our analysis consists of two cases: the case in which we try to avoid finding a bad rule, and the case in which we try to avoid overlooking a good rule. We also discuss about related works including PAC learning [5,7], Jensen and Cohen's multiple comparison [4], Agrawal et al.'s analysis of association rule discovery [1], and our previous analysis of a discovered rule based on simultaneous reliability evaluation

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 365–377, 2001.

[10]. In the rest of this paper, technical terms and symbols of referenced papers are unified to those of this paper.

2 Rule Discovery Problem

2.1 Rule

Let a data set contain n examples each of which is expressed by b discrete attributes and a class attribute. Typically rule discovery assumes no specific class attribute unlike classification. However, for the sake of formalization, we consider a rule which predicts a specific class attribute to be true.

Let a value v assignment $A = v$ to an attribute A be an atom. In this paper, we regard a given data set as a result of sampling with replacement from a true data set. We call the probability of examples each of which satisfies a propositional logical formula f the true probability $\Pr(f)$ of f. Similarly, an estimated probability which is obtained from a given data set for $\Pr(f)$ is represented by $\widehat{\Pr}(f)$. Note that $\widehat{\Pr}(f)$ can be calculated by the Laplace estimate or simply by the ratio of examples which satisfy f in the data set. We employ the latter method in this paper.

A rule r is represented as follows with a premise y which represents a propositional formula of atoms, and a conclusion x which represents a true assignment to the class attribute.

$$r : y \rightarrow x$$

An intuitive interpretation of r is that many examples satisfy y and those examples are likely to satisfy x with high probability. We define $\Pr(y)$ and $\Pr(x|y)$ as the generality and the accuracy of r respectively. Similarly, we call $\widehat{\Pr}(y)$ and $\widehat{\Pr}(x|y)$ the estimated generality and the estimated accuracy of r respectively.

2.2 Related Classes of Rules

This section presents several classes of rules which are related to ours. A probabilistic if-then rule [9] is defined as follows, where y_i represents a single atom.

$$y_1 \wedge y_2 \wedge \cdots \wedge y_K \rightarrow x$$

In [10], a probabilistic if-then rule is called a conjunction rule, and this paper follows this paraphrasing. A conjunction rule can be regarded as a special case of our rule: the premise is restricted to either a single atom or a conjunction of atoms.

Since a premise of a conjunction rule is represented by a combination of atoms, the number $|R|$ of possible conjunction rules is typically huge. The following gives $|R|$, where a data set contains b attributes and each of these attributes can have one of a values.

$$|R| = (a+1)^b - 1 \tag{1}$$

This formula can be explained by the fact that each of b attributes can either have one of a values or be excluded from the premise. A typical value for $|R|$ is huge: for example, $|R| = 3{,}486{,}784{,}400$ for a data set of 20 binary attributes. A realistic measure would be to restrict the number of atoms allowed in the premise to at most K. The possible number $|R_K|$, in this case, is given as follows.

$$|R_K| = \sum_{i=1}^{K} a^i \binom{b}{i} \tag{2}$$

Note that (1) can be also derived by settling $K = b$ in (2) and considering the binary coefficients.

In association rule discovery [1], a data set is restricted to a transactional data set which consists of binary attributes. A true assignment to a binary attribute is called an item. Let an itemset be either a single item or a conjunction of items. An association rule, in its original form, consists of a premise and a conclusion each of which is represented by an itemset. In our framework of section 2.1, an association rule can be regarded as a special case of our rule: the premise is restricted to either a single atom or a conjunction of atoms, and only the value "true" is allowed. The cases of $|R|$ and $|R_K|$ for association rule discovery are obtained by settling $a = 1$ in (1) and (2).

2.3 Discovery Problem

In this paper, the objective of a user is to obtain, with high probability $1-\delta$, a rule of which generality and accuracy are no smaller than $1-\zeta$ and $1-\epsilon$ respectively. Typically multiple rules are obtained in rule discovery, but we restrict ourselves to single-rule discovery for the sake of analysis.

$$\begin{aligned} &\text{Objective : Find } y \rightarrow x \text{ which satisfies} \\ &\qquad \Pr\left[\Pr(y) \geq 1-\zeta,\ \Pr(x|y) \geq 1-\epsilon\right] \geq 1-\delta \\ &\qquad \text{where } \zeta,\ \epsilon,\ \delta\ >\ 0 \end{aligned} \tag{3}$$

A discovery algorithm to be analyzed obtains a rule of which generality and accuracy are no smaller than user-given thresholds θ_{S} and θ_{F} respectively. As stated above, since a given data set is a result of sampling from a true data set, the user employs thresholds $\theta_{\mathrm{S}} \neq 1-\zeta!$ $\theta_{\mathrm{F}} \neq 1-\epsilon$ in applying the algorithm.

$$\begin{aligned} &\text{Algorithm : Find } y \rightarrow x \text{ which satisfies} \\ &\qquad \widehat{\Pr}(y) \geq \theta_{\mathrm{S}},\ \widehat{\Pr}(x|y) \geq \theta_{\mathrm{F}} \end{aligned} \tag{4}$$

An interesting problem here is to bound the required number m of examples to accomplish (3) under (4). This problem can be named as PAGA (Probably Approximately General and Accurate) discovery after the well-known PAC (Probably Approximately Correct) learning [5,7], and can be regarded as a foundation of data mining.

3 Case 1: Exclusion of a Bad Rule

In this section, we derive a lower bound of the number of examples for the problem defined in the previous section. An assumed condition is to avoid finding a bad rule. This condition can be considered as important in several domains where reliability represents a crucial concern.

3.1 Preliminaries

First we introduce preliminaries which are needed in subsequent analyses. If the domain of a probabilistic variable X is $\{0, 1, \cdots, m\}$ and the probability distribution of the variable is represented as follows, X is said to follow a binary distribution [3].

$$\begin{aligned} \Pr(X = k) &= \mathrm{B}(k; m, p) \\ &= \binom{m}{k} p^k (1-p)^{m-k} \end{aligned} \tag{5}$$

where p represents a constant $0 < p < 1$ and $k = 0, 1, \cdots, m$. The Chernoff bound states that the following holds for an arbitrary constant $a > p$ [1].

$$\Pr(X > am) < \exp[-2m(a-p)^2] \tag{6}$$

3.2 Theoretical Analysis

From (3), a bad rule $r_{\mathrm{b}} : y \to x$ satisfies

$$\Pr(y) < 1 - \zeta \text{ or } \Pr(x|y) < 1 - \epsilon. \tag{7}$$

Since we assume, in this section, that we avoid finding a bad rule, the employed thresholds for generality and accuracy are relatively large. This assumption together with (3) and (4) necessitate the following.

$$\theta_{\mathrm{S}} > 1 - \zeta \text{ and } \theta_{\mathrm{F}} > 1 - \epsilon \tag{8}$$

From (7) and (8),

$$\theta_{\mathrm{S}} > \Pr(y) \text{ or } \theta_{\mathrm{F}} > \Pr(x|y). \tag{9}$$

Since $r_{\mathrm{b}} : y \to x$ is discovered,

$$\widehat{\Pr}(y) \geq \theta_{\mathrm{S}} \text{ and } \widehat{\Pr}(x|y) \geq \theta_{\mathrm{F}}. \tag{10}$$

Let the number of examples in the given data set be m. If and only if y and xy are satisfied by at least $\lceil m\theta_{\mathrm{S}} \rceil$ and $\lceil m\widehat{\Pr}(y)\theta_{\mathrm{F}} \rceil$ examples respectively in the

data set, r_{b} happens to be discovered. Since each of the numbers of examples which satisfy y and xy follows a binary distribution,

$$\begin{aligned}
&\Pr\left(r_{\mathrm{b}} \text{ discovered}\right) \\
&\leq \mathrm{MAX}\left[\sum_{k=\lceil m\theta_{\mathrm{S}}\rceil}^{m} \mathrm{B}(k;m,\Pr(y)), \sum_{k=\lceil m\widehat{\Pr}(y)\theta_{\mathrm{F}}\rceil}^{m\widehat{\Pr}(y)} \mathrm{B}(k;m\widehat{\Pr}(y),\Pr(x|y))\right] \quad (11)\\
&< \mathrm{MAX}\left\{\exp\left[-2m\left(\frac{\lceil m\theta_{\mathrm{S}}\rceil}{m}-\Pr(y)\right)^2\right],\right. \\
&\qquad \left.\exp\left[-2m\widehat{\Pr}(y)\left(\frac{\lceil m\widehat{\Pr}(y)\theta_{\mathrm{F}}\rceil}{m\widehat{\Pr}(y)}-\Pr(x|y)\right)^2\right]\right\} \quad (12)\\
&< \mathrm{MAX}\left\{\exp\left[-2m(\theta_{\mathrm{S}}-1+\zeta)^2\right], \exp\left[-2m\theta_{\mathrm{S}}(\theta_{\mathrm{F}}-1+\epsilon)^2\right]\right\}. \quad (13)
\end{aligned}$$

Note that, in (11), we consider separately the case in which a bad rule $r_{\mathrm{b}1}$ in terms of generality is discovered and the case in which a bad rule $r_{\mathrm{b}2}$ in terms of accuracy is discovered. The first and second terms correspond to the left inequality and the right inequality of (7) respectively. Since $\Pr(r_{\mathrm{b}1})$ and $\Pr(r_{\mathrm{b}2})$ are unknown, we bound $\Pr(r_{\mathrm{b}}$ discovered) by $\mathrm{MAX}[\Pr(r_{\mathrm{b}1}$ discovered), $\Pr(r_{\mathrm{b}2}$ discovered)]. In (12), the Chernoff bound (6) is employed from (9). Finally in (13), we employ (7) and the left inequality of (10).

Let the set of all possible rules and the set of all bad rules be R and R_{b} respectively, and let the cardinality of a set S be $|S|$. The probability of discovering a bad rule satisfies the following inequalities.

$$\begin{aligned}
&\Pr\left(R_{\mathrm{b}} \text{ contains a discovered rule}\right) \\
&< |R_{\mathrm{b}}|\mathrm{MAX}\left\{\exp\left[-2m(\theta_{\mathrm{S}}-1+\zeta)^2\right], \exp\left[-2m\theta_{\mathrm{S}}(\theta_{\mathrm{F}}-1+\epsilon)^2\right]\right\} \quad (14)\\
&\leq |R|\mathrm{MAX}\left\{\exp\left[-2m(\theta_{\mathrm{S}}-1+\zeta)^2\right], \exp\left[-2m\theta_{\mathrm{S}}(\theta_{\mathrm{F}}-1+\epsilon)^2\right]\right\} \quad (15)
\end{aligned}$$

Note that we allow to count multiple times the cases in which several bad rules satisfy the discovery condition in (14), and (15) uses $|R| \geq |R_{\mathrm{b}}|$. Our objective (3) requires the following with respect to a sufficiently small δ.

$$|R|\mathrm{MAX}\left\{\exp\left[-2m(\theta_{\mathrm{S}}-1+\zeta)^2\right], \exp\left[-2m\theta_{\mathrm{S}}(\theta_{\mathrm{F}}-1+\epsilon)^2\right]\right\} \leq \delta \quad (16)$$

We obtain a lower bound of the number m of examples for discovery in which finding a bad rule is avoided with a high probability.

$$m \geq \frac{\ln\left(\frac{|R|}{\delta}\right)}{2\mathrm{MIN}\left[\left(\theta_{\mathrm{S}}-1+\zeta\right)^2, \theta_{\mathrm{S}}\left(\theta_{\mathrm{F}}-1+\epsilon\right)^2\right]} \quad (17)$$

The above inequality describes influence of each parameter to the minimum number of examples quantitatively. As we have seen in section 2.2, $|R|$ is typically large and is thus important even if its influence is tolerated by a logarithmic

function. The second most important factors are $\theta_{\mathrm{S}} - 1 + \zeta$ and $\theta_{\mathrm{F}} - 1 + \epsilon$. Since they influence the lower bound of m by the inverse of their squares, they can be problematic when they are small. Since each of these terms represents the difference of a threshold and the user-expected value, $\theta_{\mathrm{S}} - 1 + \zeta$ and $\theta_{\mathrm{F}} - 1 + \epsilon$ can be named as the margin of generality and the margin of accuracy respectively. In a typical setting of rule discovery, we can assume $\theta_{\mathrm{S}} = 0.1$, and we assume that $(\theta_{\mathrm{S}} - 1 + \zeta) = 10^{-1}$ or 10^{-2}. We also assume that $(\theta_{\mathrm{F}} - 1 + \epsilon) = 10^{-1}$ or 10^{-2}. Under these assumptions, the denominator is either $2 * 10^{-3}$ or $2 * 10^{-5}$. Finally, δ can be considered as a moderately important factor in a typical situation $\delta =$ 0.01 - 0.05 since it appears only as a denominator of $|R|$.

3.3 Application to Conjunction Rule Discovery

From (1) and (17), the lower bound of the number m of examples is given as follows if we restrict the discovered rule to a conjunction rule.

$$m \geq \frac{\ln\left[(a+1)^b - 1\right] + \ln\left(\frac{1}{\delta}\right)}{2\mathrm{MIN}\left[\left(\theta_{\mathrm{S}} - 1 + \zeta\right)^2, \theta_{\mathrm{S}}\left(\theta_{\mathrm{F}} - 1 + \epsilon\right)^2\right]} \tag{18}$$

Note that settling $a = 1$ gives the case of association rule discovery.

Firstly, $\ln(1/\delta)$ can be typically ignored when $\delta = 0.01$ - 0.05 from $\ln[(a + 1)^b - 1] \gg \ln(1/\delta)$, thus the lower bound of m is approximately proportional to b. Secondly, since the number a of possible values for an attribute only affects the right-hand side through a logarithmic function, a is typically not so important as b and margins of generality and accuracy. We show, in figure 1, a plot of the lower bound against $\mathrm{MIN}[(\theta_{\mathrm{S}} - 1 + \zeta)^2, \theta_{\mathrm{S}}(\theta_{\mathrm{F}} - 1 + \epsilon)^2]$ for $b = 10^2, 10^3, 10^4$, where we settled $a = 2$ and $\delta = 0.05$. Note that each of the x axis and the y axis is represented by a logarithmic scale.

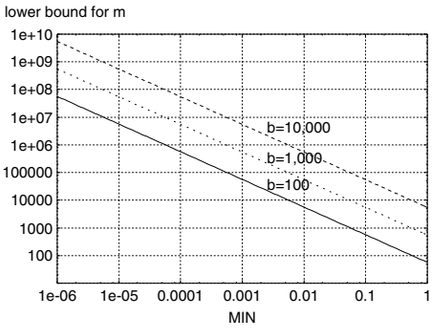


Fig. 1. Minimum number of examples needed for conjunction rule discovery without finding a bad rule. In the figure, MIN represents $\mathrm{MIN}[(\theta_{\mathrm{S}} - 1 + \zeta)^2, \theta_{\mathrm{S}}(\theta_{\mathrm{F}} - 1 + \epsilon)^2]$.

We discuss about the lower bound of the number of examples for a typical setting with figure 1. The examples described in section 3.2 state MIN$[(\theta_{\mathrm{S}} - 1 + \zeta)^2, \theta_{\mathrm{S}}(\theta_{\mathrm{F}} - 1 + \epsilon)^2] = 10^{-3}$ or 10^{-5}. For these cases, the lower bound is approximately $5.6 * 10^4$ - $5.6 * 10^6$ or $5.6 * 10^6$ - $5.6 * 10^8$ for $b = 10^2$ - 10^4. These results indicate that the required number of examples for successful discovery can be prohibitively large for small margins. Note that large margins represent large thresholds, and no rules are usually discovered for large thresholds. A realistic and effective measure to this problem would be to adjust thresholds according to a discovery process such as [11]. It should be anyway noted that our analyses in this paper correspond to the worst case, and the required number of examples in a real discovery problem can be much smaller than those mentioned above.

From (2) and (17), the lower bound of the number m of examples is given as follows if we restrict the discovered rule to a conjunction rule with at most K atoms in its premise.

$$m \geq \frac{\ln\left[\sum_{i=1}^{K} a^i \binom{b}{i}\right] + \ln\left(\frac{1}{\delta}\right)}{2\mathrm{MIN}\left[\left(\theta_{\mathrm{S}} - 1 + \zeta\right)^2, \theta_{\mathrm{S}}\left(\theta_{\mathrm{F}} - 1 + \epsilon\right)^2\right]} \tag{19}$$

Note that settling $a = 1$ gives the case of association rule discovery.

Similarly as we did in figure 1, we show, in figure 2, two plots of the lower bound for $a = 2$ and $\delta = 0.05$. The left plot represents a case in which we varied $b = 10^2,\ 10^3,\ 10^4$ under $K = 2$, and in the right plot we varied $K = 1,\ 2,\ 3,\ 4,\ 100\ (= b)$ under $b = 10^2$.

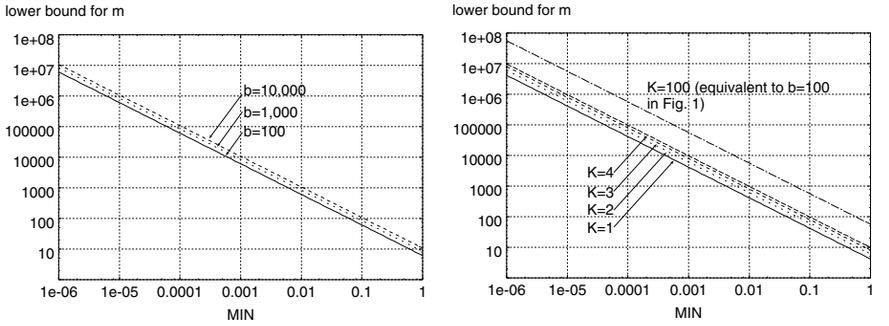


Fig. 2. Minimum number of examples needed for conjunction rule discovery without finding a bad rule, where at most K atoms are allowed in the premise. The left and right plots assume $K = 2$ and $b = 100$ respectively.

From the left plot, we see that the influence of b is relatively small for $K = 2$. On the other hand, the right plot of figure 2 shows that, for $K \leq 4$, the minimum required number of examples is smaller by approximately an order of magnitude

than the case of considering all conjunction rules ($K = b = 100$). It is widely accepted that a rule with a short premise exhibits high readability, and the above results suggest that they are also attractive in terms of the required number of examples.

4 Case 2: Inclusion of a Good Rule

In this section, we derive another lower bound of the number of examples for the problem defined in section 2.3. An assumed condition is to avoid overlooking a good rule. This condition can be considered as important in several domains where possibility is considered as highly important.

From (3), a good rule $r_{\mathrm{g}} : y \to x$ satisfies

$$\Pr(y) \geq 1 - \zeta \text{ and } \Pr(x|y) \geq 1 - \epsilon. \tag{20}$$

Since we assume, in this section, that we avoid overlooking a good rule, the employed thresholds for generality and accuracy are relatively small. This assumption together with (3) and (4) necessitate the following.

$$\theta_{\mathrm{S}} < 1 - \zeta \text{ and } \theta_{\mathrm{F}} < 1 - \epsilon \tag{21}$$

From (20) and (21),

$$\theta_{\mathrm{S}} < \Pr(y) \text{ and } \theta_{\mathrm{F}} < \Pr(x|y). \tag{22}$$

Let the number of examples in the given data set be m. If and only if y is satisfied by at most $\lceil m\theta_{\mathrm{S}} \rceil - 1$ examples or xy is satisfied by at most $\lceil m\widehat{\Pr}(y)\theta_{\mathrm{F}} \rceil - 1$ examples in the data set, r_{g} happens to be undiscovered. Since each of the numbers of examples which satisfy y and xy follows a binary distribution,

$$\begin{aligned}
&\Pr\,(r_{\mathrm{g}} \text{ undiscovered}) \\
&\leq \sum_{k=0}^{\lceil m\theta_{\mathrm{S}} \rceil - 1} \mathrm{B}(k; m, \Pr(y)) + \sum_{k=0}^{\lceil m\widehat{\Pr}(y)\theta_{\mathrm{F}} \rceil - 1} \mathrm{B}(k; m\widehat{\Pr}(y), \Pr(x|y)) \\
&\quad - \left[\sum_{k=0}^{\lceil m\theta_{\mathrm{S}} \rceil - 1} \mathrm{B}(k; m, \Pr(y)) \right] \left[\sum_{k=0}^{\lceil m\widehat{\Pr}(y)\theta_{\mathrm{F}} \rceil - 1} \mathrm{B}(k; m\widehat{\Pr}(y), \Pr(x|y)) \right] \qquad (23) \\
&< \sum_{k=0}^{\lceil m\theta_{\mathrm{S}} \rceil - 1} \mathrm{B}(k; m, \Pr(y)) + \sum_{k=0}^{\lceil m\widehat{\Pr}(y)\theta_{\mathrm{F}} \rceil - 1} \mathrm{B}(k; m\widehat{\Pr}(y), \Pr(x|y)) \qquad (24) \\
&= \sum_{k=m-\lceil m\theta_{\mathrm{S}} \rceil + 1}^{m} \mathrm{B}(k; m, 1 - \Pr(y)) \\
&\quad + \sum_{k=m\widehat{\Pr}(y)-\lceil m\widehat{\Pr}(y)\theta_{\mathrm{F}} \rceil + 1}^{m\widehat{\Pr}(y)} \mathrm{B}(k; m\widehat{\Pr}(y), 1 - \Pr(x|y)) \qquad (25)
\end{aligned}$$

$$< \exp\left[-2m\left(\frac{m-\lceil m\theta_{\mathrm{S}}\rceil+1}{m}-1+\Pr(y)\right)^2\right]$$

$$+\exp\left[-2m\widehat{\Pr}(y)\left(\frac{m\widehat{\Pr}(y)-\lceil m\widehat{\Pr}(y)\theta_{\mathrm{F}}\rceil+1}{m\widehat{\Pr}(y)}-1+\Pr(x|y)\right)^2\right] \quad (26)$$

$$\leq \exp\left\{-2m\left[(-\theta_{\mathrm{S}}+1-\zeta)^2+\theta_{\mathrm{S}}(-\theta_{\mathrm{F}}+1-\epsilon)^2\right]\right\}. \quad (27)$$

Note that we consider separately the cases in which the generality and the accuracy of a good rule are below the respective thresholds in (23). In (24), we allow to double-count the probability of the simultaneous occurrence of these cases, and assume that r_{g} is undiscovered due to apparently low accuracy in the second term. Note that (25) corresponds to replacement of p by $1-p$ in (5). In (26), the Chernoff bound (6) is employed from (22). Finally in (27), we employ (20) and $\widehat{\Pr}(y) \geq \theta_{\mathrm{S}}$. Note that the last inequality holds in the second term since r_{g} is undiscovered due to apparently low accuracy.

Similarly to section 3.2, the following can be obtained as a lower bound of the number m of examples for discovery in which overlooking a good rule is avoided with a high probability.

$$m \geq \frac{\ln\left(\frac{|R|}{\delta}\right)}{2\left[(-\theta_{\mathrm{S}}+1-\zeta)^2+\theta_{\mathrm{S}}\,(-\theta_{\mathrm{F}}+1-\epsilon)^2\right]} \quad (28)$$

Note that (28) can be obtained by substituting in (17) the add-sum of the two terms $(-\theta_{\mathrm{S}}+1-\zeta)^2,\ \theta_{\mathrm{S}}(-\theta_{\mathrm{F}}+1-\epsilon)^2$ in the denominator for the minimum of these two terms. In section 3, we had to bound $\Pr(r_{\mathrm{b}}$ discovered) by MAX[$\Pr(r_{\mathrm{b1}}$ discovered), $\Pr(r_{\mathrm{b2}}$ discovered)] since $\Pr(r_{\mathrm{b1}})$ and $\Pr(r_{\mathrm{b2}})$ are unknown. In this section, on the other hand, we can directly calculate Pr[(r_{g} undiscovered due to apprarently low generality) $\cup$ (r_{g} undiscovered due to apprarently low accuracy)]. The substitution is due to this difference.

Under this condition, similar discussions as section 3.2 and 3.3 hold for (28). Note that large margins ($1-\zeta-\theta_{\mathrm{S}}$ and $1-\epsilon-\theta_{\mathrm{F}}$ in this case) represent small thresholds in this case, and small thresholds typically result in a large number of candidates of the discovered rule to be inspected. The automatic adjustment of thresholds [11] can be also a realistic measure for this problem.

5 Discussions on Related Topics

5.1 PAC Learning

PAC learning represents a worst-case analysis for classification, and has numerous excellent results. Our results in section 3.2 can be considered as an extension to a preliminary version of PAC learning [7]. First, a classifier ignores generality since it predicts the class attribute for all examples. This is the reason $\Pr(y)$!

$\widehat{\Pr(y)}$! θ_{S} are not considered in [7]. Actually, the objective of learning in [7] is represented as a simplification of (3) as follows, where h represents a classifier.

$$\begin{aligned}\text{Objective}: &\text{ Find } h \text{ which satisfies}\\ &\Pr\left[\Pr(h \text{ correctly predicts } x) \geq 1-\epsilon\right] \geq 1-\delta \\ &\text{where } \epsilon,\ \delta\ >\ 0\end{aligned} \tag{29}$$

Next, [7] assumes that a classification algorithm returns a classifier which is consistent to all training examples. This corresponds to assuming $\theta_{\mathrm{F}} = 1$.

To sum up, compared to our study, [7] ignores the case of learning a classifier with low generality and the case of learning a classifier which is inconsistent to the training examples. In this case, application of the Chernoff bound can be skipped, and for a bad classifier h_{b}, we obtain $\Pr(h_{\mathrm{b}} \text{ learned}) = (1-\epsilon)^m$. In [7], a lower bound of the required number of examples m is given by the following, where H represents a set of all classifiers.

$$m \geq \frac{\ln\left(\frac{|H|}{\delta}\right)}{\epsilon} \tag{30}$$

Note that (30) resembles to (17): it only ignores generality ($\theta_{\mathrm{S}} = 1$ and no ζ), assumes $\theta_{\mathrm{F}} = 1$, and omits the squares in ϵ^2 and 2 in the denominator. The last omissions are due to skipping the Chernoff bound.

5.2 Jensen and Cohen's Multiple Comparison

Jensen and Cohen's multiple comparison [4] proposes a prudent view of classification. Its essential point can be stated as a probabilistic explanation that the more candidates of classifiers are inspected in a learning algorithm, the smaller accuracy is exhibited by the obtained classifier. The multiple comparison provides a comprehensive unified view of several studies including overfitting [8] and oversearching [6], and [4] also proposes several realistic measures.

Since this study deals with classification as PAC learning, it ignores generality. This corresponds to considering only the second term in (11). Since [4] considers the case of $\theta_{\mathrm{F}} < 1$, it provides a more realistic framework to learning than [7]. The multiple comparison differs from our study in that it directly calculates, based on a binary distribution without using the Chernoff bound, the probability for a bad classifier to satisfy at least $\lceil m\theta_{\mathrm{F}} \rceil$ examples. Moreover, they calculate exactly the probability that no bad classifier is learned while we, in (14), allow counting multiples times the cases in which more than one bad rules satisfy the discovery condition. Let the set of all bad classifiers be H_{b}, the probability in [4] is given by the following.

$$\Pr(H_{\mathrm{b}} \text{ contains a learned classifier}) = 1 - [1 - \Pr(h_{\mathrm{b}} \text{ learned})]^{|H|} \tag{31}$$

Pursuing strictness in calculation can be considered as a double-edged sword. Jensen and Cohen give no analytical solutions to the required number of examples for successful learning. We attribute this reason to the fact that resolving

(31) for m is relatively difficult. We have employed several approximations in our theoretical analyses, and these were necessary to bound m analytically. Another difference between [4] and our analyses is rather philosophical: while they are pessimistic about classification, we are realistic about rule discovery. The study in [4] emphasizes that $|H|$ is huge, and demonstrates various examples in which it is difficult to avoid learning a bad classifier. We also recognize that $|R|$ is huge, but bounds the required number of examples m analytically with respect to $|R|$.

5.3 Theoretical Analysis of Association Rule Discovery

Analyses of association rule discovery [1] are threefold: a lower bound of the number of queries under the use of a database system, the expected number of itemsets each of which is satisfied by at least a required number of examples in a random data set, and the number of examples satisfied by an itemset in a sampled data set. The third analysis is highly related to our study in that both of the two deal with the case of sampling m examples from a true data set in rule discovery.

The analysis provides a specification of the Chernoff bound (6), where X is regarded as $m\widehat{\Pr}(f)$ for an itemset f. It first regards the right-hand side $\exp[-2m(a-p)^2]$ as the upper bound of the probability for $\widehat{\Pr}(f)$ to deviate at least $a-p$ from its value p $(=\Pr(f))$ in the true data set. Next, it gives several examples of values for $a-p$ and δ in $\exp[-2m(a-p)^2]=\delta$, and represents the corresponding values of m in a table.

The discovery algorithm employed in [1] first obtains, by an algorithm called Apriori, a set of imtemsets f each of which satisfies $\widehat{\Pr}(f) \geq \theta_{\mathrm{S}}$. Then, it generates a set of association rules from this set. One of the motivations of the above analysis was to reduce the run-time of Apriori by the use of a sampled data set. Due to this motivation, [1] ignores accuracy unlike our study. Moreover, since it considers a single association rule, the study fails to relate the size of a discovery problem to the number of examples needed for successful discovery.

5.4 Simultaneous Reliability Evaluation of a Discovered Rule

Simultaneous reliability evaluation of a discovered rule [10] also deals with the case of sampling m examples from a true data set in rule discovery as in section 5.3 and our study. Unlike the analysis in section 5.3, this study considers both generality and accuracy.

The objective considered in [10] is identical to ours, and is represented by (3). Let $\overline{x}$ represent the negation of x. The analysis fixes m and employs neither θ_{S} nor θ_{F}. It assumes that $(m\Pr(xy), m\Pr(\overline{x}y))$ follows a two-dimensional normal distribution, and obtains the exact condition for accomplishing the objective analytically. This is a different framework from ours: we use a discovery algorithm with fixed thresholds θ_{S}, θ_{F} in (4) and bound the number m of sampled examples. The problem dealt in [10] can be reduced to the problem of deriving

and analyzing two tangent lines of an ellipse, and applying Lagrange's multiplier method gives the following analytical solutions.

$$\left(1-\beta(\delta)\sqrt{\frac{1-\widehat{\Pr}(y)}{n\widehat{\Pr}(y)}}\right)\widehat{\Pr}(y)\geq 1-\zeta \tag{32}$$

$$\left(1-\beta(\delta)\sqrt{\frac{\widehat{\Pr}(\overline{x},y)}{\widehat{\Pr}(x,y)\{(n+\beta(\delta)^2)\widehat{\Pr}(y)-\beta(\delta)^2\}}}\right)\widehat{\Pr}(x|y)\geq 1-\epsilon \tag{33}$$

Here $\beta(\delta)$ represents a positive constant which defines the size of a $1-\delta$ confidence region i.e. the ellipse for $(m\Pr(xy), m\Pr(\overline{x}y))$, and can be obtained by a simple numerical integration. Note that (32) and (33) represent conditions for generality and accuracy respectively. Each of them states that the corresponding estimated probability multiplied by a coefficient which is related to the size of the confidence region is no smaller than the corresponding user-expected value ($1-\zeta$ or $1-\epsilon$).

Since the study [10] assumes a specific distribution to the simultaneous occurrence of random variables, it does not fall in the category of worst-case analysis. Similarly to the analysis in section 5.3, the study fails to relate the size of a discovery problem to the number of examples needed for successful discovery.

6 Conclusions

The main contribution of this paper is threefold. 1) We formalized a worst-case analysis of rule discovery. The proposed framework employs thresholds $\theta_{\rm S}$, $\theta_{\rm F}$ for generality and accuracy which are different from user-expected values $1-\zeta$! $1-\epsilon$ respectively. We considered the case in which we try to avoid finding a bad rule, and the case in which we try to avoid overlooking a good rule. 2) We derived a lower bound of the number m of required examples. By using probabilistic formalization and appropriate approximations, two lower bounds are obtained for the aforementioned two cases. Quantitative analysis of one of the lower bounds revealed that the total number $|R|$ of rules, the margin $\theta_{\rm S}-1+\zeta$ for generality, and the margin $\theta_{\rm F}-1+\epsilon$ for accuracy are important. 3) We analyzed one of the lower bounds for a set of specific problems of conjunction rule discovery. Various useful conclusions are obtained by inspecting the lower bound for a set of typical settings.

The contribution of 1) represents that this paper has provided, in rule discovery, a framework which corresponds to PAC learning. This framework can be named as PAGA (Probably Approximately General and Accurate) discovery. PAGA discovery can be regarded as promising as a theoretical foundation of active mining, which requests new examples in a discovery process. The contributions of 2) and 3) suggest various useful policies in applying various rule algorithms in practice. Such policies include sampling/extension of a data set and modification of the class of discovered rules. We can safely conclude that

our comprehension to rule discovery has deepened with these contributions and discussions in section 5. Ongoing work focuses on analyses of more realistic algorithms, especially an algorithm which discovers multiple rules with various conclusions.

Acknowledgement

We are grateful to Setsuo Arikawa for enabling us to initiate this study by suggesting us to pursue the relationship of one of our previous studies and PAC learning. This work was partially supported by the grant-in-aid for scientific research on priority area "Active Mining" from the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo: Fast Discovery of Association Rules, *Advances in Knowledge Discovery and Data Mining*, pp. 307–328, AAAI/MIT Press, Menlo Park, Calif. (1996).
2. U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth: "From Data Mining to Knowledge Discovery: An Overview", *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, pp. 1–34, Menlo Park, Calif. (1996).
3. W. Feller: *An Introduction to Probability Theory and Its Applications*, John Wiley & Sons, New York (1957).
4. D. D. Jensen and P. R. Cohen: "Multiple Comparisons in Induction Algorithms", *Machine Learning*, Vol. 38, No. 3, pp. 309–338 (2000).
5. M. J. Kearns and U. V. Vazirani: *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, Mass. (1994).
6. J. R. Quinlan and R. Cameron-Jones: "Oversearching and Layered Search in Empirical Learning", *Proc. Fourteenth Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, pp. 1019–1024 (1995).
7. S. Russel and P. Norvig: *Artificial Intelligence, a Modern Approach*, pp. 552–558, Prentice Hall, Upper Saddle River, N. J. (1995).
8. C. Schaffer: "Overfitting Avoidance as Bias", *Machine Learning*, Vol. 10, No. 2, pp. 153–178 (1993).
9. P. Smyth and R. M. Goodman: "An Information Theoretic Approach to Rule Induction from Databases", *IEEE Trans. Knowledge and Data Eng.*, Vol. 4, No. 4, pp. 301–316 (1992).
10. E. Suzuki: "Simultaneous Reliability Evaluation of Generality and Accuracy for Rule Discovery in Databases", *Proc. Fourth Int'l Conf. on Knowledge Discovery and Data Mining (KDD)*, pp. 339–343 (1998).
11. E. Suzuki: "Scheduled Discovery of Exception Rules", *Discovery Science, LNAI 1721 (DS)*, pp. 184–195, Springer-Verlag (1999).

Mining Semi-structured Data by Path Expressions

Katsuaki Taniguchi[1], Hiroshi Sakamoto[1], Hiroki Arimura[1,2], Shinichi Shimozono[3], and Setsuo Arikawa[1]

[1] Department of Informatics, Kyushu University, Fukuoka 812-8581, Japan, {k-tani, hiroshi, arim, arikawa}@i.kyushu-u.ac.jp,
[2] PRESTO, Japan Science Technology Co., Japan,
[3] Department of Artificial Intelligence, Kyushu Institute of Technology Iizuka 820-8502, Japan, sin@ai.kyutech.ac.jp

Abstract. A new data model for filtering semi-structured texts is presented. Given positive and negative examples of HTML pages labeled by a labelling function, the HTML pages are divided into a set of paths using the XML parser. A path is a sequence of *element nodes* and *text nodes* such that a text node appears in only the tail of the path. The labels of an element node and a text node are called a *tag* and a *text*, respectively. The goal of a mining algorithm is to find an interesting pattern, called *association path*, which is a pair of a tag-sequence t and a word-sequence w represented by the *word-association pattern* [1]. An association path (t, w) *agrees with* a labelling function on a path p if t is a subsequence of the tag-sequence of p and w matches with the text of p iff p is in a positive example. The importance of such an associate path α is measured by the *agreement* of a labelling function on given data, i.e., the number of paths on which α agrees with the labelling function. We present a mining algorithm for this problem and show the efficiency of this model by experiments.

1 Introduction

In the *information extraction*, it is one of the central problems in Web mining to detect the occurrences or the regions of useful texts. In case of the Web data, this problem is particularly difficult because we can not represent a rich logical structure by the limited tags of the HTML. The framework of *wrapper induction* introduced by Kushmerick [13] is a new approach to handle this difficulty. The most interesting result of his study is to show the effectiveness and efficiency of simple wrappers with string delimiters in the information extraction tasks. Together with his work, we can find other extracting models, for example, in [8, 9,10,11,15,17].

The target class, called *HTML pages*, of the wrapper induction model is restricted such that a page is defined by a finite repetition of a sequence of

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 378–388, 2001.

attributes. The attributes are the data which an algorithm has to extract. In a learning model, a learning algorithm takes an input of labeled examples such that the labels indicate whether they are *positive* data or *negative* data. The strategy is useful to learn a *concept* for the wrapper class.

However, in case that a concept class is hard to learn by a small number of examples, the model may not be effective. This difficulty is critical in the point of implementation since the labelling examples are actually made by human inspection. Thus, we would like to present a mining model to decide which portion of a given data is important and an automatic process to construct a large labelling sample.

The aim of this paper is to find rules for filtering semi-structured texts according to users interests. An HTML/XML file can be considered as an ordered labeled tree. We assume that each node is either an *element node* and a *text node*. Each node has two types of labels called the *name* and *value*. An element node corresponds to a tag. The name of the node is the tag name like `<HTML>`, `<br>`, and `<a>`, and the value of the node is empty. A text node corresponds to a portion of a plain text in an HTML and the name is the reserved string $\sharp Text$, respectively. The value of a text node is the text.

A filtering rule is a sequence $s = \langle \alpha_1, \ldots, \alpha_k, \beta \rangle$, where α_i is a tag name, β is a *word-association pattern* [1] which is a string consisting of several words and the wild card $*$. A word-association pattern *matches with a string* if there is a possible substitution for all $*$. Given the s and a semi-structured text, using an XML parser, we can easily construct the tree structure and decompose the tree into the set P of paths. Each path contains at most one text node in the tail. The semantics of the filtering rule s for P is defined as follows. For each $p \in P$, s matches with p if $\alpha_1, \ldots, \alpha_k$ is a subsequence of the sequence of tag names of p and the tail of p is a text node such that β matches with the value of the node.

Such a filtering rule is considered as a simple decision tree to extract texts from paths in HTML trees. Each α_i represents a test on a node. Unless the test is failed, we continue the test to the next test α_{i+1}. Finally, the value of the text node is extracted according to the pattern β. In other words, this rule is a pair of *tag patterns* and association patterns $\langle \alpha, \beta \rangle$, where a tag pattern is a sequence $\alpha = (\alpha_1, \ldots, \alpha_k)$ of tag names such that these tags frequently appear in positive examples together with the association pattern. Such a filtering rule is called an *association path* in this paper. We can use this notion for a measure to decide the importance of keyword in a text. We show the efficiency of the association paths by experiments.

This paper is organized as follows. In Section 2, we define the data model for HTML pages, HTML trees, and path expressions. In Section 3, we review the definition of the word-association pattern in [1] and formulate the mining problem, called Association Path problem, of this paper. Next we describe a mining algorithm which finds an association path for given a large collection of HTML pages. In Section 4, we show several experimental results. In the first experiment, the set of positive examples is a collection of HTML texts containing a keyword "TSP" and the set of negatives is that containing "NP". These key-

words mean the *travelling sealsman problem* and *NP-optimization problem* on the computational complexity theory, respectively. The aim is to find an association path to characterize the notion TSP comparing to NP. In this experiment, the algorithm found some interesting association paths. For the next experiment, we choose the keyword "DNA" for positive examples. Compared to the first result, the algorithm found few interesting paths. In Section 5, we conclude this study.

2 The Data Model

In this section, we introduce the data model considered in this paper. First, we begin with the notations used in this paper. $\mathbb{N}$ denotes the set of all nonnegative integers. An *alphabet* Σ is a set of finite symbols. A finite sequence $\langle a_1, \ldots, a_n \rangle$ of elements in Σ is called *string* and it is denoted by $w = a_1 \cdots a_n$ for short. The *empty string* of length zero is ε. The set of all strings is denoted by Σ^* and let $\Sigma^+ = \Sigma^* - \{\varepsilon\}$. For string w, if $w = \alpha\beta\gamma$, then the strings α and γ are called a *prefix* and a *suffix* of w, respectively. For a string s, we denote by $s[i]$ with $1 \leq i \leq |s|$ the i-th symbol of s, where $|s|$ is the length of s.

For an HTML page, the HTML trees are the ordered node-labeled trees defined as follows. For each tree T, the set of all nodes of T is a finite subset of $\mathbb{N}$, where the 0 is the root. A node is called a *leaf* if it has no child and otherwise called an *internal node*. If nodes $n, m \in \mathbb{N}$ have the same parent, then n and m are *siblings*. A sequence $\langle n_1, \ldots, n_k \rangle$ of nodes of T is called a *path* if n_1 is the root and n_i is the parent of n_{i+1} for all $i = 1, \ldots, k-1$. For a path $p = \langle n_1, \ldots, n_k \rangle$, the number k is called *the length of* p and the node n_k is called *the tail of* p.

With each node n, the pair $NL(n) = \langle N(n), V(n) \rangle$, called *the node label of* n, is attached, where $N(n)$ and $V(n)$ are strings called the *node name* and *node value*, respectively. If $N(n) \in \Sigma^+$ and $V(n) = \varepsilon$, then the node n is called the *element node* and the string $N(n)$ is called the *tag*. If $N(n) = \sharp Text$ for the reserved string $\sharp Text$ and $V(n) \in \Sigma^+$, then n is called the *text node* and the $V(n)$ called the *text value*. We assume that every node $n \in \mathbb{N}$ is categorized to the element node or text node.

If a page P contains a beginning tag of the form <*tag*> and P contains no ending tag corresponding to it. Then, the tag <*tag*> is called an *empty tag in* P. If a page P contains a string of the form $t_1 \cdot w \cdot t_2$ such that t_1, t_2 are either beginning or ending tags and w is a string not containing any tag, then the string w is called a *text in* P.

An HTML file is called a *page*. A page P corresponds to an ordered labeled tree. For the simplicity, we assume that the P contains no comments, which is any string beginning the string <!- and ending the string ->.

Definition 1. For a page P, we define the HTML tree P_t recursively as follows.

1. If P contains an empty tag of the form <*tag*>, then P_t has the element node n such that it is a leaf of P, $N(n) = tag$, and $V(n) = \varepsilon$.

2. If P contains a text w, then P_t has the text node n such that it is a leaf P, $N(n) = \sharp Text$, $V(n) = w$.
3. If P contains a string of the form `<tag>`s`</tag>` for a string $s \in \Sigma^*$, then P_t has the subtree $n(n_1, \ldots, n_k)$, where $N(n) = tag$, $V(n) = \varepsilon$ and $n_1, \ldots, n_k$ are the roots of the trees $t_1, \ldots, t_k$ which are obtained from the w by recursively applying the above 1, 2 and 3.

Next we define the functions to get the node names, node values, and HTML attributes from given nodes and HTML trees defined above. These functions are useful to explain the algorithms in the next section. These functions return the values indicated below and return *null* if such values do not exist.

- *Parent*(n): The parent of the node $n \in I\!N$.
- *ChildNodes*(n): The sequence of all children of n.
- *Name*(n): The node name $N(n)$ of n.
- *Value*(n): The concatenation $V(n_1) \cdots V(n_k)$ for the leaves $n_1, \ldots, n_k$ of the subtree rooted at n in the left-to-right order.

Recall that $V(n)$ is not empty only if n is text node. Thus, *Value*(n) is equal to the concatenation of values of all text nodes below n. Let P_t be an HTML tree for a page P and let $N = \{0, \ldots, n\}$ be the set of nodes in P_t. For nodes $i, j \in N$, if there is a sequence $p_{i,j} = \langle i_1, \ldots, i_k \rangle$ of nodes in N such that $i_1 = i$, $i_k = j$, and $i_\ell = Parent(i_{\ell+1})$ for all $1 \leq \ell \leq k-1$, then the $p_{i,j}$ is called the *path* from i to j. If i is the root, then $p_{i,j}$ is denoted by p_j for short. For each path $p = \langle i_1, \ldots, i_k \rangle$ of P_t, we also define the following useful notations.

- *Name*(p): The sequence $\langle Name(i_1), \ldots, Name(i_k) \rangle$.
- *Value*(p): $V(n_k)$.

Definition 2. Let P_t be an HTML tree over the set N of nodes. Let $p = \langle i_1, \ldots, i_n \rangle$ be a path of P_t and let $Name_t = \{Name(n) \mid n \in N\}$. A sequence $\alpha = \langle name_1, \ldots, name_m \rangle$, $(name_i \in Name_t)$ is called a *path expression over* $Name_t$. It is called that *the* α *matches with the* p if there exists a subsequence $j_1, \ldots, j_m$ of p such that $Name(j_\ell) = name_\ell$ for all $1 \leq \ell \leq m$.

In the next section, we define a measure of the matching of the path expressions with the paths of HTML trees. We also define the finding problem of a path expression to maximize the measure.

3 Mining HTML Texts

In this section we first define the problem to find an expression, called an *association pattern*, for filtering semistructured texts. The pattern is a pair of a *word-association pattern* and a path expression. The semantics of the patterns is defined by the matching semantics of the word-association patterns and the path expressions.

3.1 The Problem

A *word-association pattern* [1] π over Σ is a pair $\pi = (p_1, \ldots, p_d; k)$ of a finite sequence of strings in Σ^* and a parameter k called *proximity* which is either a nonnegative integer or infinity. A word-association pattern π *matches* a string $s \in \Sigma^*$ if there exists a sequence $i_1, \ldots, i_d$ of integers such that every p_j in π occurs in s at the position i_j and $0 \leq i_{j+1} - i_j \leq k$ for all $1 \leq j < d-1$. The notion (d, k)*-pattern* refers to a d-word k-proximity word-association pattern $(p_1, \ldots, p_d; k)$.

Let $S = \{s_1, \ldots, s_m\}$ be a finite set of strings Σ^* and let ψ be a labeling function $\psi : S \to \{0, 1\}$. Then, for a string $s \in S$, we say that a word-association pattern π *agrees with* ψ *on* s if π matches s iff $\psi(s) = 1$.

Given (Σ, S, ψ, d, k) of an alphabet Σ, a finite set $S \subset \Sigma^*$ of strings, a labeling function $\psi : S \to \{0, 1\}$, and positive integers d and k, the problem Max Agreement by (d, k)-Pattern [1] is to find a (d, k)-pattern π such that it maximizes the *agreement* of ψ, i.e., the number of strings in S on which π agrees with ψ.

Definition 3. An *association path* is an expression of the form $\alpha\#\pi$, where the α is a path expression such that its tail is $\sharp Text$, the π is a word-association pattern, and the $\#$ is the special symbol not belonging to any α and π. Let $p = \alpha\#\pi$ be an association path and p' be a path in a tree. It is said that *the* p *matches the* p' if α matches p' and π matches $\mathit{Value}(p')$.

For a finite set T of HTML trees, let

$$\mathit{Text}_T = \{\langle \mathit{Name}(p), \mathit{Value}(p)\rangle \mid p \text{ is a path of } t \in T,\ \mathit{Value}(p) \neq \varepsilon\}$$

The intuitive meaning of p appearing in Text_T is a path p of an HTML tree such that the tail of p is a text node. Let Name_T be the set of $\mathit{Name}(p)$ and let Value_T be the set of $\mathit{Value}(p)$ in Text_T.

Definition 4. Association Path
An instance is $(\Sigma, \mathit{Text}_T, \psi, d, k)$ of an alphabet Σ, a set Text_T of pairs for a finite set T of HTML trees, a labeling function $\psi : \mathit{Value}_T \to \{0, 1\}$, and positive integers d, k. A solution is an association path $\alpha\#\pi$. The string π is a (d, k)-pattern for a solution of the max agreement problem for input $(\Sigma, \mathit{Value}_T, \psi, d, k)$. The string α is a (d, k)-pattern for a solution of the max agreement problem for input $(\Sigma, \mathit{Name}_T, \psi', d, k)$ such that where ψ is defined by $\psi'(\mathit{Name}(p)) = 1$ iff $\psi(\mathit{Value}(p)) = 1$. The goal of the problem is to maximize the sum of the agreements of ψ and ψ' over all association paths $\alpha\#\pi$.

3.2 The Algorithm

To find association paths, the data of HTML texts are transformed to path expressions as follows. Given a large set S of HTML texts, it is divided into two disjoint sets S_1 and S_2 by a labeling function. The labeling function is considered as a keyword or phrase by a user, i.e., any text in S is labeled by 1 if it contains

the keyword and labeled by 0 otherwise. Next all texts in S_1 and S_2 are parsed to HTML trees and let *Pos* be the set all paths from S_1 and *Neg* be the set of all paths from S_2. Fig. 3.2 shows the process of our algorithm briefly.

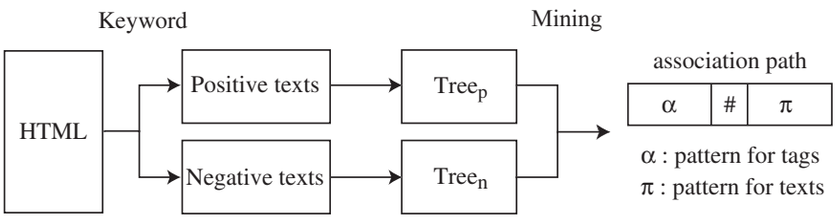


Fig. 1. The process of mining algorithm.

Algorithm **Path-Find**(Σ, *Text*, ψ, d, k)
/* Input: a set of HTML pages P over Σ, a labeling function ψ, non negative integers d, k */
/* Output: a solution of ASSOCIATION PATH for the input*/

1. Let P_1 be the set of all pages in P labeled by 1 and let $P_2 = (P - P_1)$. For the set T_1 of HTML trees of P_1, compute the set *Pos* of all paths of trees in P_1 and the set *Neg* of all paths of all trees in P_2.
2. Let $Pos = \{p_i \mid 1 \leq i \leq m\}$ and $Neg = \{q_j \mid 1 \leq j \leq n\}$ $(m, n \geq 0)$. Compute the sets $Name_{Pos} = \{Name(p) \mid p \in Pos\}$, $Value_{Pos} = \{Value(p) \mid p \in Pos\}$, $Name_{Neg} = \{Name(q) \mid q \in Neg\}$, and $Value_{Neg} = \{Value(q) \mid q \in Neg\}$.
3. Find a (d, k)-pattern π of the max agreement problem for $\langle Value_{Pos}, Value_{Neg}\rangle$, and find a (d, k)-pattern α of the max agreement problem for $\langle Name_{Pos}, Name_{Neg}\rangle$.
4. Output the pattern $\alpha\#\pi$ which maximizes the sum of the agreement of α and π.

We estimate the running time of the **Path-Find**. This algorithm finds an association path for only the paths whose tails are the text nodes, i.e., the paths of the form $p = \langle n_1, \ldots, n_k\rangle$, the n_i $(1 \leq i \leq k-1)$ is an element node and the n_k is a text node. Thus, for such paths p, we regard the mining problem as the problem to find two phrases α from the strings $Name(p)$ and β from the strings $Value(p)$ for constant parameters d of the number of phrases of texts and k of the distance of phrases.

If the maximum number of phrases in a pattern is bounded by a constant d then the max agreement problem for (d, k)-patterns is solvable by *Enumerate-Scan* algorithm [19], a modification of a naive generate-and-test algorithm, in

$O(n^{d+1})$ time and $O(n^d)$ scans although it is still too slow to apply real world problems.

Adopting the framework of optimized pattern discovery, we have developed an efficient algorithm, called *Split-Merge* [1], that finds all the optimal patterns for the class of (d, k)-patterns for various statistical measures including the classification error and information entropy.

The algorithm quickly searches the hypothesis space using dynamic reconstruction of the content index, called a *suffix array* with combining several techniques from computational geometry and string algorithms.

We showed that the Split-Merge algorithm runs in *almost linear time in average*, more precisely in $O(k^{d-1}N(\log N)^{d+1})$ time using $O(k^{d-1}N)$ space for nearly random texts of size N [1]. We also show that the problem to find one of the best phrase patterns with arbitrarily many strings is MAX SNP-hard [1]. Thus, we see that there is no efficient approximation algorithm with arbitrary small error for the problem when the number d of phrases is unbounded.

4 Experimental Results

In this section, we show the experimental results. The text data is a collection from the *ResearchIndex* [1] which is a scientific literature digital library. A positive data is the set *Pos* of HTML pages containing the keyword "TSP" and a negative data is the set *Neg* of HTML pages containing the keyword "NP". The set *Neg* consists of many topics of computational complexity problems and *Pos* is concerned with one of the most popular NP-hard problems *Travelling Salesman Problem* not properly contained in *Neg*. The aim of this experiment is to find an association path which characterizes TSP with NP.

By this experiment on the collection of 8.4MB, the algorithm **Path-Find** finds the best 600 patterns at the entropy measure in seconds for $d = 2$ and three minutes for $d = 3$ with $k = 10$ words using 200 mega-bytes of main memory on IBM PC (PentiumIII 600 MHz, gcc++ on Windows98). The result obtained by our algorithm is shown in Fig. 1.

Our system found several interesting association paths which may be difficult for human users to find by inspection. Fig. 1 consists of some association paths whose tag sequences contains `<i>` tag. This means that the phrases, e.g., 'local search' and 'euclidean tsp', are emphasized by the tag. Thus we consider these phrases to be interesting. In fact these phrases are remarkable by the following reasons.

The phrase 'local search' in Rank 171 indicates the *local search heuristics* for TSP such as [14]. In this path, the tag `<i>` and `<font>` (font style and size) in the left hand side indicates the importance of the phrase `<local search>` in the right hand side. The phrase 'tsp and other' in Rank 276 is a substring of the title of the outstanding paper written by Arora [2] in 1996 on the approximation algorithm for Euclidean TSP. The *euclidean* graph is an important *geometric*

[1] http://citeseer.nj.nec.com/

Rank	Association path $\alpha\#\pi$
5	< i font p body html> # <tsp >
38	< i font p body html> # <for the >
90	< i font p body html> # <the tsp >
171	< i font p body html> # <local search >
213	< i font p body html> # <traveling >
276	< i font p body html> # <tsp and other >
394	< i font p body html> # <euclidean tsp >
455	< i font p body html> # <other geometric problems >
552	< i > # <approximation schemes for euclidean >

Fig. 2. The association paths found in the experiments, which characterize the Web pages on the TSP problem from these on NP-optimization problm. The parameters are $(2, 10)$ for (d, k), where α is a path and π is a phrase.

structure to construct an *approximation* algorithm for TSP. These keywords appear in Rank 394, 455, and 552, respectively.

Next we examine the same text data by the association pattern algorithm [1] and compare the resulting phrases with our result. The list of 400 phrases found by the association pattern algorithm is partially presented in Fig. 1. As is shown in this list, almost phrases are trivial except 'local search'.

0 <the ><tsp >	10 <tsp ><and >
1 <<font ><tsp >	11 <local search ><the >
2 <for ><tsp >	12 <the ><salesman >
3 <and ><tsp >	13 <and ><np >
4 <tsp ><in >	14 <the ><np >
5 <tsp ><of >	15 <and ><local search >
6 <tsp ><the >	16 <tsp ><a >
7 <of ><tsp >	17 <np ><the >
8 <= ><for the >	18 <for ><traveling >
9 <<font ><for the >	19 <local ><the >

Fig. 3. The top 20 patterns of 400 association patterns found by the algorithm in [1]. The parameter is $(2, 10)$ for (d, k).

Moreover it is difficult to recognize the importance of 'local search' by this result only because it is an ordinary phrase in computer science. On the other hand, we show all phrases containing the emphasis tag <i> in Fig. 1. Compared with Fig. 1, we can comfirm that none of the important phrases 'local search', 'euclidean tsp',

'geometric', and 'approximation' appears in the list of Fig. 1. Thus we conclude the effectiveness of our algorithm on this examination.

145	<<i>the ><tsp >
212	<<i>the ><solutions for tsp >
213	<<i>the ><for tsp >
214	<<i>the traveling ><tsp >
215	<<i>the traveling ><solutions for tsp >
216	<<i>the traveling ><for tsp >
240	<<i>the ><computational solutions for tsp >
241	<<i>the traveling ><computational solutions for tsp >
256	<<font ><<i>the traveling >

Fig. 4. All association patterns containing <i> tag found by the algorithm in [1]. The parameter is $(2, 10)$ for (d, k).

Finally, we show other experimental results. The positive sample is a set of HTML pages containing the keyword "DNA" and the negative sample is the same to above experiment. By this experiment on the collection of 9.3MB, the algorithm finds the best 600 patterns at the entropy measure in seconds for $d = 2$ with $k = 10$. The result is shown in Figure 3. Our system found few of association paths containing interesting keywords like "sequence" "computer" and "molecular". In this result, several paths containing the *anchor tag*. Unfortunately, interesting keywords are not found in such paths.

Rank	Association path $\alpha\#\pi$
23	<a > # <dna >
136	<i font p body html > # <dna sequences >
199	<i font p body html > # <molecular >
360	<i font p body html > # <computer >
395	<a > # <computation >
444	<a body html > # <computing >

Fig. 5. Other result of the experiments for the DNA from NP-optimization problem. The parameters are also $(2, 10)$ for (d, k), where α is a path and π is a phrase.

5 Conclusion

We introduced a new method for mining from HTML texts and present an algorithm to find an association path which is a pair of association patterns over tag sequences and text sequences. By experiments on HTML data of scientific literature, the algorithm found interesting association paths from positive and negative examples on the traveling salesman problem and the other NP optimization problems.

Acknowledgments

The authors would be grateful to the anonymous referees for their careful reading of the draft and uesful comments. Shinichi Shimozono thanks Miho Matsui for the suggestive discussions and observations obtained while supervising her graduation thesis.

References

1. Shimozono, S., Arimura, H., and Arikawa, S. Efficient discovery of optimal word-association patterns in large text databases. *New Generation Computing* 18:49-60, 2000.
2. Arora, S. Polynomial-time approximation schemes for Euclidean TSP and other geometric problems. *Proc. 37th IEEE Symposium on Foundations of Computer Science*, 2-12, 1996.
3. Abiteboul, S., Buneman, P., and Suciu, D. Data on the Web: From relations to semistructured data and XML, Morgan Kaufmann, San Francisco, CA, 2000.
4. Angluin, D. Queries and concept learning. *Machine Learning* 2:319–342, 1988.
5. Buneman, P., Davidson, S., Hillebrand, G., and Suciu, D. A query language and optimization techniques for unstructured data. *University of Pennsylvania, Computer and Information Science Department, Technical Report MS-CIS* 96-09, 1996.
6. Cohen, W. W. and Fan, W. Learning Page-Independent Heuristics for Extracting Data from Web Pages, *Proc. WWW-99.* 1999.
7. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S. Learning to construct knowledge bases from the World Wide Web, *Artificial Intelligence* 118:69–113, 2000.
8. Freitag, D. Information extraction from HTML: Application of a general machine learning approach. *Proc. the 15th National Conference on Artificial Intelligence*, 517-523, 1998
9. Grieser, G., Jantke, K. P., Lange, S., and Thomas, B. A unifying approach to HTML wrapper representation and learning, *Proc. the 3rd International Conference, DS2000*, Lecture Notes in Artificial Intelligence 1967:50–64, 2000.
10. Hammer, J., Garcia-Molina, H., Cho, J., and Crespo, A. Extracting semistructured information from the Web. *Proc. Workshop on Management of Semistructured Data*, 18–25, 1997.
11. Hsu, C.-N. Initial results on wrapping semistructured web pages with finite-state transducers and contextual rules. *Proc. 1998 Workshop on AI and Information Integration*, 66–73, 1998.
12. Kamada, T. Compact HTML for small information appliances. *W3C NOTE 09-Feb-1998.* `www.w3.org/TR/1998/NOTE-compactHTML-19980209`, 1998.

13. Kushmerick, N. Wrapper induction: efficiency and expressiveness. *Artificial Intelligence* 118:15–68, 2000.
14. Lin, S., and Kernighan, B. W. An effective heuristic algorithm for the travelling salesman problem. *Operations Research* 21:498-516, 1973.
15. Muslea, I., Minton, S., and Knoblock, C. A. Wrapper induction for semistructured, web-based information sources. *Proc. Conference on Automated Learning and Discovery*, 1998.
16. Sakamoto, H., Arimura, H., and Arikawa, S. Identification of tree translation rules from examples. *Proc. the 5th International Colloquium on Grammatical Inference*, LNAI 1891:241–255, 2000.
17. Thomas, B. Anti-unification based learning of T-Wrappers for information extraction, *Proc. AAAI Workshop on Machine Learning for IE*, 15–20, AAAI, 1999.
18. Valiant, L. G. A theory of the learnable. *Comm. ACM* 27:1134–1142, 1984.
19. Wang, J. T., Chirn, G. W., Marr, T. G., Shapiro, B., Shasha, D., and Zhang, K. Combinatorial pattern discovery for scientific data: Some preliminary results. *Proc. SIGMOD'94*, 115-125, 1994.

Theory Revision in Equation Discovery

Ljupčo Todorovski and Sašo Džeroski

Department of Intelligent Systems, Jožef Stefan Institute
Jamova 39, 0.50 Ljubljana, Slovenia
Ljupco.Todorovski@ijs.si, Saso.Dzeroski@ijs.si

Abstract. State of the art equation discovery systems start the discovery process from scratch, rather than from an initial hypothesis in the space of equations. On the other hand, theory revision systems start from a given theory as an initial hypothesis and use new examples to improve its quality. Two quality criteria are usually used in theory revision systems. The first is the accuracy of the theory on new examples and the second is the minimality of change of the original theory. In this paper, we formulate the problem of theory revision in the context of equation discovery. Moreover, we propose a theory revision method suitable for use with the equation discovery system LAGRAMGE. The accuracy of the revised theory and the minimality of theory change are considered. The use of the method is illustrated on the problem of improving an existing equation based model of the net production of carbon in the Earth ecosystem. Experiments show that small changes in the model parameters and structure considerably improve the accuracy of the model.

1 Introduction

Most of the existing equation discovery systems make use of a very limited portion of the theoretical knowledge available in the domain of interest. Usually, the domain knowledge is used to constrain the search space of possible equations to the equations that make sense from the point of view of the domain experts. One of the aspects of the domain knowledge that is usually neglected by the equation discovery systems are the existing models in the domain. Rather than starting the search with an existing equation based model, equation discovery systems always start their search from scratch. In contrast with them, theory revision systems [9,3] start with an existing model and use heuristic search to revise the model in order to improve its fit to observational data.

Most of the work on theory revision systems is on the revision of theories in propositional and first-order logic [9]. In this paper, we propose a flexible grammar based framework for theory revision in equation discovery. The existing initial model is transformed to a grammar, and alternative productions are used to define a space of possible revised equation models. The grammar based equation discovery system LAGRAMGE [6] is then used to search through the space of revised models and find the one that fits observational data best. The use of the proposed framework is illustrated on revising an equation based earth-science model of the net production of carbon in the Earth ecosystem.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 389–400, 2001.

The paper is organized as follows. The following section give a brief introduction to grammar based equation discovery. Typical approaches to revision of theories in propositional and first-order logic are briefly reviewed in Section 3. The grammar based framework for theory revision in equation discovery is presented in Section 4. Section 5 presents the experiments with revising the earth-science equation model. The last section summarizes the paper, discusses related work and gives direction for further work.

2 Equation Discovery

Equation discovery is the area of machine learning that develops methods for automated discovery of quantitative laws, expressed in the form of equations, in collections of measured data [1]. Equation discovery systems heuristically search through a subset of the space of all possible equations and try to find the equation which fits the measured data best.

Different equation discovery systems explore different spaces of possible equations. Early equation discovery systems used pre-defined (built-in) spaces that were small enough to allow effective heuristic (or exhaustive) search. However, this approach does not allow the user of the equation discovery system to tailor the space of possible equation to the domain of interest. On the other hand, recent equation discovery systems use different approaches to allow the user to restrict the space of the possible equations. In equation discovery systems that are based on genetic programming, the user is allowed to specify a set of algebraic operators that can be used. A similar approach has been used in the EF [10] equation discovery system. The equation discovery system SDS [7] effectively uses user provided scale-type information about the dimensions of the system variables and is capable of discovering complex equations from noisy data.

Finally, the equation discovery system LAGRAMGE [6] allows the user to specify the space of possible equations using a context free grammar. Note that grammars are a more general and powerful mechanism for tailoring the space of the equations to the domain of use than the ones used in SDS [7] and EF [10]. In the rest of this section we will describe this grammar based approach to equation discovery used in LAGRAMGE.

2.1 Grammar-Based Equation Discovery

The problem of grammar based equation discovery can be formalized as follows.
Given:

- a set of variables $V = v_1, v_2, \ldots, v_n$ of the observed system, including a target dependent variable $v_d \in V$;
- a grammar G; and
- a table M of observations (measured values) of the system variables.

Find a model E in the form of one or more algebraic or differential equations defining the target variable v_d that:

1. is derived by the grammar G; and

2. minimizes the discrepancy between the observed values of the target variable v_d and the values of v_d obtained with simulating the model.

An example of a grammar for equation discovery is given in Table 1. The grammar contains a set of two nonterminal symbols {`P_Vdiff, Vdiff`}, with a set of productions attached to each of them, and a set of three terminal symbols {`v1, v2, const[0:1]`}. The semantics of the terminal and nonterminal symbols in the grammar are explained below.

There are two types of terminal symbols used in the grammars for equation discovery. The first group is used to denote the variables of the observed system (`v1` and `v2` in the example grammar from Table 1). Another group of terminal symbols of the form `const[l:h]` is used to denote the constant parameter in the equation model whose value has to be fitted against the observational data from M. A constraint `[l:h]` specifies that the value of the constant parameter should be within the interval $l \leq v \leq h$.

Table 1. An example of a grammar for equation discovery defining the space of polynomials of a single variable $vdiff = v_1 - v_2$.

```
P_Vdiff -> const[0:1]
P_Vdiff -> const[0:1] + (P_Vdiff) * (Vdiff)
Vdiff -> v1 - v2
```

The nonterminal symbol `Vdiff` defines an intermediate variable which is the difference between two system variables `v1` and `v2`. This is done with the single production for the nonterminal symbol `Vdiff`. The other nonterminal symbol `P_Vdiff` is used to build polynomials of an arbitrary degree.

2.2 Lagramge

The equation discovery system Lagramge applies heuristic (or exhaustive) search through the space of models generated using user provided grammar G. The values constant parameters (terminal symbols `const`) in the generated models are fitted against input data M using standard non-linear constrained optimization method. After fitting the values of the constant parameters tho model is evaluated according to the sum of squared errors (SSE heuristic function [6]), i.e., the differences between observed values of the target variable v_d and the values of v_d calculated by the model. Alternative MDL heuristic function that takes into account the complexity of the model can be also used [6].

3 Theory Revision

The problem of theory revision can be defined as follows: **Given** an imperfect domain theory in the form of classification rules and a set of classified examples,

find an approximately minimal syntactic revision of the domain theory that correctly classifies all of the examples.

A representative system that addresses this problem is EITHER [3]. EITHER refines propositional Horn-clause theories using a suite of abductive, deductive and inductive techniques. Deduction is used to identify the problems with the domain theory, while abduction and induction are used to correct them. The problem of theory revision has received a lot of attention in the field of inductive logic programming [2], where a number of approaches have been developed for revising theories in the form of first-order Horn clause theories. For an overview, we refer the reader to [9].

Two kinds of problems are encountered within imperfect domain theories: over-generality occurs when an example is classified into a class other than the correct one, while over-specificity occurs when an example cannot be proven to belong to the correct class. Note that a single example can be misclassified both ways at the same time. Overly general rules are either specialized by adding new conditions to their antecedents or are deleted from the knowledge base. Problems of over-specificity are solved by generalizing the antecedents of existing rules, e.g., by removing conditions from them, or by the induction of new rules.

4 Grammar-Based Theory Revision of Equation Models

4.1 Problem Definition

The problem of grammar based theory revision can be formalized as follows. **Given**:

- a set of variables $V = v_1, v_2, \ldots, v_n$ of the observed system, including a target dependent variable $v_d \in V$;
- an existing model E, represented as an equation(s) defining the target variable v_d. Note that this can actually be a set of (algebraic or differential) equations defining the value of the target variable v_d;
- a grammar G that derives the model E; and
- a table M of observations (measured values) of the system variables.

Find a revised model E' (equation/set of equations as above) that:

1. is derived by the grammar G;
2. minimizes the discrepancy between the observed values of the target variable v_d and the values of v_d obtained with simulating the model; and
3. differs from the existing model E as little as possible.

Items 2. and 3. above would typically appear in a formulation of a general theory revision problem, regardless of the language in which the theories are expressed. In contrast to our formulation, however, the possible changes to the initial theory would be specified in terms of revision operators that can be applied to the initial and intermediate theories. As theories are typically logical theories in theory revision settings, operators typically include addition/deletion of entire rules (propositional or first-order Horn clauses) and addition/deletion of conditions in individual rules.

4.2 From an Initial Model to a Grammar

In a typical setting of revising an existing scientific model, we would only have observational data and a model, i.e., an equation developed by scientists to explain a particular phenomenon. A grammar that would explain how this model was actually derived and provide options for alternative models is typically not available. The above is especially true for simpler models.

However, when the model (equation) is complex, it is only rarely written as a single equation defining the target variable, but rather as a set of equations defining the target variable, which typically contains equations defining intermediate variables. The latter typically define meaningful concepts in the domain of discourse. Often, alternative equations defining an intermediate variable would be possible and the modeling scientist would choose one of these: the alternatives would rarely (if ever) be documented in the model itself, but might be mentioned in a scientific article describing the derived model and the modeling process.

Table 2. Equations defining the NPPc variable in the CASA earth-science model.

$$
\begin{array}{l}
NPPc = \max(0, E \cdot IPAR) \\
\quad E = 0.389 \cdot T1 \cdot T2 \cdot W \\
\quad\quad T1 = 0.8 + 0.02 \cdot topt - 0.0005 \cdot topt^2 \\
\quad\quad T2 = 1.1814/((1 + e^{0.2 \cdot (TDIFF - 10)}) \cdot (1 + e^{0.3 \cdot (-TDIFF - 10)})) \\
\quad\quad\quad TDIFF = topt - tempc \\
\quad\quad W = 0.5 + 0.5 \cdot eet/PET \\
\quad\quad\quad PET = 1.6 \cdot (10 \cdot \max(tempc, 0)/ahi)^A \cdot pet\_tw\_m \\
\quad\quad\quad\quad A = 0.000000675 \cdot ahi^3 - 0.0000771 \cdot ahi^2 + 0.01792 \cdot ahi + 0.49239 \\
\quad IPAR = FPAR\_FAS \cdot monthly\_solar \cdot SOL\_CONV \cdot 0.5 \\
\quad\quad FPAR\_FAS = \min((SR\_FAS - 1.08)/srdiff, 0.95) \\
\quad\quad\quad SR\_FAS = (1 + fas\_ndvi/1000)/(1 - fas\_ndvi/1000) \\
\quad\quad SOL\_CONV = 0.0864 \cdot days\_per\_month
\end{array}
$$

A set of equations defining a target variable through some intermediate variables can easily be turned into a grammar, as demonstrated in Tables 2 and 3, which give an earth-science model and a grammar that derives this model only. Having the grammar in Table 3, however, enables us to specify alternative models through providing additional productions for the nonterminal symbols in the grammar. Additional productions for intermediate variables would specify alternative choices, only one of which will eventually be chosen for the final model. Observational data would be then used to select among combinations of such choices, if we apply a grammar based equation discovery system (such as LAGRAMGE) with the grammar that includes additional productions to observational data as input.

While the presented approach from the previous paragraph does take into account the initial model, it may allow for a completely different model to be

Table 3. Grammar derived from the equations for NPPc variable in the CASA earth-science model in Table 2. The grammar generates the original equations only.

```
NPPc ->     max(const[0:0], E * IPAR)
E ->        const[0.389:0.389] * T1 * T2 * W
T1 ->       const[0.8:0.8] + const[0.02:0.02] * topt
            - const[0.0005:0.0005] * topt * topt
T2 ->       const[1.1814:1.1814] / ((const[1:1] + exp(const[0.2:0.2]
            * (TDIFF - const[10:10]))) * (const[1:1]
            + exp(const[0.3:0.3] * (-TDIFF - const[10:10]))))
TDIFF ->    topt - tempc
W ->        const[0.5:0.5] + const[0.5:0.5] * eet / max(PET, const[0:0])
PET ->      const[1.6:1.6]
            * pow(const[10:10] * max(tempc, const[0:0]) / ahi, A)
            * pet_tw_m
A ->        const[0.000000675:0.000000675] * ahi * ahi * ahi
            - const[0.0000771:0.0000771] * ahi * ahi
            + const[0.01792:0.01792] * ahi + const[0.49239:0.49239]
IPAR ->     FPAR_FAS * solar * SOL_CONV * const[0.5:0.5]
FPAR_FAS -> min((SR_FAS - const[1.08:1.08]) / srdiff, const[0.95:0.95])
SR_FAS ->   (const[1:1] + fas_ndvi / const[1000:1000])
            / (const[1:1] - fas_ndvi / const[1000:1000])
SOL_CONV -> const[0.0864:0.0864] * days_per_month
```

derived, depending on whether productions for alternative definitions are provided for each of the intermediate variables. It is here that the minimal revision/change principle comes into play: among theories of similar quality (fit to the data), theories that are closer to the original theory are to be preferred. Since we are dealing with theories that are not necessarily expressed in logic (e.g., equations), only syntactic criteria of minimality of change are applicable in a straightforward fashion.

4.3 Typical Alternative Productions

Note that when an alternative production is specified for an intermediate variable, there are no restrictions (at least in principle) on these productions. For example, they can introduce new intermediate variables and productions defining them. They can also specify arbitrary functional forms (in the case of equations). However, they do have to eventually derive (in the context of the entire grammar) valid sub-expressions involving the set of terminal symbols (system variables) associated to the initial model.

A very common alternative production would replace the particular constants on the right-hand-side with generic constants, allowing the equation discovery system to re-fit them to the given observational data. In the grammar from Table 3 that change can be achieved by replacing a terminal symbol of the form `const[v:v]`, denoting a constant parameter with fixed value `v`, with a

generic symbol `const` that allows for an arbitrary value of the particular constant parameter. In our experiments with the earth-science CASA model we allow for a 100% change of the original values of the constant parameters in the initial model. This can be specified by replacing the terminal symbol `const[v:v]` with `const[0:2*v]`, where interval $[0 : 2 \cdot v]$ is equal to $[(v - 100\% \cdot v) : (v + 100\% \cdot v)]$ (a 100% relative change).

A slightly more complex alternative production would replace a particular polynomial on the right-hand-side of a production with an arbitrary polynomial of the same (intermediate) variables. For example, in the grammar from Table 3 can be replaced by a grammar, similar to the example grammar from Table 1, for generating an arbitrary polynomial of the variable *topt*.

4.4 Current Implementation

Our current implementation of the theory revision approach to equation discovery outlined above involves applying LAGRAMGE to the given observational data and a grammar specifying the possible alternative productions to be used in theory revision. The observational data are used to select a particular combination of the possible alternatives: note that these also include leaving parts of the model unchanged (as the original productions are also a part of the grammar) even if alternative productions for these exist.

We currently do not have an implementation of the minimal change preference integrated within LAGRAMGE. This however, can be achieved in a relatively straightforward manner. One of the heuristic functions used by LAGRAMGE to search the space of equations, called MDL, takes into account the degree-of-fit (sum of square errors) as well as the size of the equation model. A reasonable approach to implement a minimality of change principle would be to replace the second term in the MDL heuristic: replace the size of the equation with a distance between the current model and the initial model. The distance measure can be a distance on tree-structured terms, which would take into account the number and complexity of the alternative productions taken to derive the current equation.

5 Experiments in Revising an Earth-Science Model

We illustrate the use of the proposed framework for theory revision in equation discovery on the problem of revising one part of the earth-science CASA model [4]. The CASA model predicts annual global fluxes in trace gas production on the basis of a number of measured (observed) variables, such as surface temperatures, satellite observations of the land surface, soil properties, etc. Because the whole CASA model is a quite complex system of difference and algebraic equations, we focused on the revision of the NPPc part of CASA (CASA-NPPc), presented in Table 2, that is used to predict the monthly net production of carbon at a given location.

The values of the input variables (terminal symbols in the grammar from Table 2) were measured (and/or calculated) for 303 locations on the Earth providing a data set with 303 examples. In order to evaluate the accuracy of the model on unseen data we applied standard ten-fold leave-one-out cross validation method. The error of the original and revised models was calculated as root mean squared error defined as $\sqrt{\sum_{i=1}^{N}(NPPc_i - \hat{NPPc})_i^2/N}$, where N is number of the data points; $NPPc_i$ and $\hat{NPPc}_i$ are the observed value and the value calculated by the model, respectively.

5.1 Revisions Used in the Experiments

As described in Section 4 we first transformed the given NPPc model into a grammar (given in Table 3) that derives that model only. Furthermore, we added alternative productions to the grammar that define the space of possible revisions. We used six alternative possibilities for the revision of the NPPc model, described below.

E-c-100 : we allowed a 100% relative change of the constant parameter 0.389 in the equation defining the intermediate variable E. Therefore, we replaced the original production for nonterminal symbol `E` in the grammar with `E -> const[0:0.778] * T1 * T2 * W`, i.e., changed the constraint on the value of the constant parameter from the original `const[0.389:0.389]`, which fixes the value of the constant parameter, to `const[0:0.778]`, which allows a 100% relative change of the original value of the constant parameter ($[0 : 0.778]$ being equal to $[(0.389 - 100\% \cdot 0.389) : (0.389 + 100\% \cdot 0.389)]$).

T1-c-100, T2-c-100 : we allowed the same revisions as the one described above on the right hand sides of the productions for `T1` and `T2`.

SR_FAS-c-20 : we allowed 20% relative change of the constant parameters values in the equation defining the intermediate variable SR_FAS. The relative change of 20% was used to avoid values of the constant parameters lower than 800, which would cause singularity (division by zero) problems in the formula for calculating SR_FAS.

T1-s : we allowed the original second degree polynomial for calculation of $T1 = 0.8 + 0.02 \cdot topt - 0.0005 \cdot topt^2$ with an arbitrary polynomial of the same variable *topt*. The following alternative productions were added to the grammar from Table 3 for this purpose: `T1 -> const` and `T1 -> const + (T1) * topt`.

T2-s : the graph of the dependency between the *T2* and *TDIFF* variables shows a Gaussian-like slightly asymmetrical dependency curve. Following the fact that this kind of dependency can be approximated also with a higher degree polynomial we replaced the original `T1` production in the grammar from Table 3 with two productions (similar to the ones for **T1-s**, presented above) that define an arbitrary polynomial of the *TDIFF* variable.

In addition to these six possibilities for revising the CASA-NPPc model we also used different combinations of them.

5.2 Results of the Experiments

The results of the experiments with different alternative grammars for revision are presented in Table 4.

Table 4. Error reduction (in %) gained with revising the original CASA-NPPc model using different grammars for revision.

Grammar	Reduction of RMSE (in %)
SR_FAS-c-20	14.93
T2-c-100	13.25
T1-s	13.05
T2-s	12.90
E-c-100	12.59
T1-c-100	12.39
SR_FAS-c-20 + T2-s	15.56
SR_FAS-c-20 + T1-s	15.46
T2-c-100 + T1-s	13.92
T2-s + T1-s	13.30
SR_FAS-c-20 + T2-c-100	11.55
SR_FAS-c-20 + T2-s + T1-s + E-c-100	**16.19**
SR_FAS-c-20 + T2-s + T1-c-100 + E-c-100	15.44
SR_FAS-c-20 + T2-c-100 + T1-s + E-c-100	14.82
SR_FAS-c-20 + T2-c-100 + T1-c-100 + E-c-100	12.92

The first six rows of Table 4 shows that revising the value of the constant parameters in the equation for calculating *SR_FAS* gives the greatest improvement of the original model. The original value of the parameters (equal to 1000) defines an almost linear dependence of *SR_FAS* on observed variable *srdiff*. The revised values of the constant parameters were equal to 800 (lowest possible values), which increase the non-linearity of the dependence. Allowing lower values of the parameters in the equation gives further improvement, but singularity (division by zero) problems appear due to the range of the *srdiff* variable.

The analysis of the results of the structural revisions shows the following. **T1-s** revision cause the second-degree polynomial for calculating the *T1* variable to be replaced by a fourth degree polynomial. On the other hand, the structural revision **T2-s** reduced the complex formula for calculating *T2* with a constant value. This is a surprising result that would have to be discussed with the Earth science experts that built the CASA model.

Furthermore, we tested pairwise combinations of the six model refinements. The results are presented in the second part of the Table 4. Results show that improvements gained using individual refinement grammars do not combine additively. However, combinations do increase the improvements: maximal improvement gained with pairwise combinations is 15.56% compared with the highest improvement of 14.93% gained using individual revisions.

Finally, the results of the experiments with combining all the refinements are presented in the last four rows of Table 4. Note however, that revisions of the T1 and T2 structures (T1-s and T2-s) are mutually exclusive with the respective revisions of the T1 and T2 constants (T1-c-100 and T2-c-100). Therefore, four possible combinations are possible, the one combining the structural revisions of the *T1* and *T2* formulas and revisions of the values of the constant parameters in formulas for the *SR_FAS* and *E* gives the maximal improvement of the accuracy of 16.19%.

In sum, the presented results of the experiments show that small revisions of the CASA-NPPc model parameters and structure considerably improve the accuracy of the model, the maximal improvement being above 16%. However, Earth science experts should also evaluate the comprehensibility and acceptability of the revised models. Nevertheless, if some of the revisions generate models that do not make sense from their point of view, new alternative productions would have to be defined to reflect the experts comments, and allow only revisions that lead to acceptable models.

Note here that the most of the error reduction is gained using a fairly simple revision operator of changing the values of the constant parameters in the *SR_FAS* equation. Only minor additional reductions can be obtained by combining this revision with any of the other five revision operators described above. Therefore, this revision would probably be the optimal one from the point of view of the minimality of change criterion, discussed in Section 4.

6 Conclusions and Discussion

We have presented a general framework for the revision of theories in the form of (sets of) quantitative equations. The method is based on grammars, which can be derived from the original theory. Domain experts can focus the revision process on parts of the model and guide it by providing relevant alternative productions. In this way, the revision process can be interactive, as is quite often the case when revising theories expressed in logic.

We have applied our approach to the problem of revising an existing equation based model of the net production of carbon in the Earth ecosystem. Experimental results show that small revisions in both the values of the constant parameters and the structure of equations considerably reduce the error of the model by 16%.

Saito et al. [5] address the same task of revising scientific models in the form of equations. Their approach is based on transforming parts of the model into a neural network, training the neural network, then transforming the trained network back into an expression/equation. This indirect approach is limited to revising the parameters or form of one equation in the model at a time. It also requires some handcrafting to encode the equations as a neural network – the authors state that "the need to to translate the existing CASA model into a declarative form that our discovery system can manipulate" is a challenge to their approach.

Our approach allows for a straightforward representation of existing scientific models as grammars, which can then be directly manipulated and used to perform theory revision. The transition from the initial model to a grammar is so straightforward that we consider automating this process as one of the topics for immediate further work. Revisions to several equations of the original model may be considered simultaneously, as illustrated by the experiments performed.

Whigham and Recknagel [8] also consider the specific task of revising an existing model for predicting chlorophyll-a by using measured data. They use a genetic algorithm to calibrate the equation parameters. They also use a grammar based genetic programming approach to revise the structure of two sub-parts (one at a time) of the initial model. A most general grammar that can derive an arbitrary expression using the allowed arithmetic operators and functions was used for each of the two sub-parts.

Unlike this paper, Whigham and Recknagel [8] do not present a general framework for the revision of quantitative scientific models. Their approach is similar to ours in that they use grammars to specify possible revisions. However, the grammars they use are too general to provide much information about the domain at hand. Also, they do not consider the notion of minimality of revision and genetic programming typically produces very large expressions without a simplicity bias.

As already mentioned, an immediate topic for further work is to automate the grammar generation from the initial model. Another challenge is to provide the domain experts an interactive tool for testing out different alternatives for revision. Furthermore, integrating the minimality of change criterion in LAGRAMGE is also an open issue. Minimal description length (MDL) heuristics in LAGRAMGE can be adapted to take into account the distance between the current and the initial equation model. Finally, we plan to apply the proposed framework to the task of revision of other portions of the CASA model as well as revision of other equation based environmental models.

Acknowledgments

We thank Christopher Potter, Steven Klooster and Alicia Torregrosa from NASA-Ames Research Center for making available both the CASA model and the relevant data set.

References

1. P. Langley, H. A. Simon, G. L. Bradshaw, and J. M. Żythow. *Scientific Discovery.* MIT Press, Cambridge, MA, 1987.
2. N. Lavrac and Sašo Džeroski. *Inductive Logic Programming: Techniques and Applications.* Ellis Horwood, Chichester, 1994. Freely available at http://www-ai.ijs.si/SasoDzeroski/ILPBook/.
3. D. Ourston and R. J. Mooney. Theory refinement combining analytical and empirical methods. *Artificial Intelligence*, 66:273–309, 1994.

4. C. S. Potter and S.A. Klooster. Interannual variability in soil trace gas (CO2, N2O, NO) uxes and analysis of controllers on regional to global scales. *Global Biogeochemical Cycles*, 12:621–635, 1998.
5. K. Saito, P. Langley, and T. Grenager. The computational revision of quantitative scientific models. 2001. Submitted to Discovery Science conference.
6. L. Todorovski and S. Džeroski. Declarative bias in equation discovery. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 376–384, Nashville, MA, 1997. Morgan Kaufmann.
7. T. Washio and H. Motoda. Discovering admissible models of complex systems based on scale-types and identity constraints. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, volume 2, pages 810–817, Nogoya, Japan, 1997. Morgan Kaufmann.
8. P. A. Whigham and F. Recknagel. Predicting chlorophyll-a in freshwater lakes by hybridising process-based models and genetic algorithms. In *Book of Abstracts of the Second International Conference on Applications of Machine Learning to Ecological Modeling*. Adelaide University, 2000.
9. S. Wrobel. First order theory refinement. In L. De Raedt, editor, *Advances in Inductive Logic Programming*, pages 14–33. IOS Press, 1996.
10. R. Zembowicz and J. M. Żytkow. Discovery of equations: Experimental evaluation of convergence. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 70–75, San Jose, CA, 1992. Morgan Kaufmann.

Simplified Training Algorithms for Hierarchical Hidden Markov Models

Nobuhisa Ueda[1,2] and Taisuke Sato[1,2]

[1] Dept. of Computer Science, Tokyo Institute of Technology
[2] CREST, JST
2-12-2 Ookayama Meguro-ku Tokyo Japan 152-8552
ueda@mi.cs.titech.ac.jp, sato@cs.titech.ac.jp

Abstract. We present a simplified EM algorithm and an approximate algorithm for training hierarchical hidden Markov models (HHMMs), an extension of hidden Markov models. The EM algorithm we present is proved to increase the likelihood of training sentences at each iteration unlike the existing algorithm called the generalized Baum-Welch algorithm. The approximate algorithm is applicable to tasks like robot navigation in which we observe sentences and train parameters simultaneously. These algorithms and their derivations are simplified by making use of stochastic context-free grammars.

1 Introduction

Hidden Markov models (HMMs) are a class of statistical language models to capture and predict uncertain phenomena from data, and have succeeded in numerous applications including speech recognition [12], and computational biology [8]. To describe more complex models, various extensions of HMMs have been proposed recently such as Input-Output HMMs [3], factorial HMMs [7], hierarchical HMMs (HHMMs) [6], and maximum entropy Markov models [11].

HHMMs were proposed to efficiently describe global dependencies over sentences (data) by incorporating hierarchical structures of sentences. To discover hierarchical structures from data, they have been applied to practical tasks such as recognition of cursive handwriting [6] and robot navigation [14]. HHMMs, however, have two issues: One is that efficiency of HHMMs has not been compared to that of stochastic context-free grammars (SCFGs) yet, though HHMMs are claimed as a simpler alternative to SCFGs. The other is an Expectation-Maximization (EM) algorithm for HHMMs called the generalized Baum-Welch algorithm [6]. We found some faults with the algorithm, and fixed them. But even after fixing them, it did *not* always increase the likelihood of training sentences, which should not happen for any EM algorithm. As using incorrect training algorithms, one might be led to discover false knowledge by inaccurate parameters.

In this paper, we prove contrary to the claim about HHMMs in literature [6] that HHMMs are efficiently representable with SCFGs. We also derive a simplified EM algorithm for HHMMs. Thanks to simplicity of the EM algorithm, we can also derive an approximate algorithm for training HHMMs.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 401–415, 2001.

2 Preliminaries

2.1 Hierarchical Hidden Markov Models

We review hierarchical hidden Markov models (HHMMs) [6]. Notation we use is different in part from the original one [6] for notational convenience. Let $Q = \{q_1, \ldots, q_n\}$ be a set of states, and $\Sigma = \{\sigma_1, \ldots, \sigma_m\}$ a set of symbols. $O = \{o^1, \ldots, o^v\} \subseteq \Sigma^+$ denotes a set of training sentences, and $o_s^u \cdots o_t^u$ $(1 \leq u \leq v)$ a part of a training sentence o^u from the s-th symbol to the t-th symbol.

An HHMM has three types of state, *internal states*, *production states*, and *end states*. In an HHMM, there is a unique internal state q_1 called the *initial state*, and q_1 has a "submodel." The submodel is composed of either at least one production state, or one end state and at least one internal state. Every internal state in a submodel has a unique individual submodel recursively. An internal state q_i and a state q_j are called a *parent state* of q_j and a *substate* of q_i, respectively, if a submodel of q_i contains q_j. An internal state q_i is said to be a *neighbor state* of a state q_j if q_i and q_j are in the same submodel. q_i^{end} denotes an end state which is a neighbor state of q_i.

Recursiveness of submodels forms a tree structure in an HHMM. In the tree, leaves, nodes, and the root correspond to production states, submodels, and a submodel containing the initial state, respectively. The depth of a state q_i is defined as the depth of a submodel containing q_i in the tree. Let D denote the maximum depth of states in an HHMM. Without loss of generality, we assume $i < j$ if q_i and q_j are at depth d and d' $(d < d')$ respectively.

In each state q_i, depending on the type of q_i, three types of transition called *vertical transitions*, *horizontal transitions*, and *forced transitions* can occur. First suppose we are in an internal state q_i. A vertical transition to q_j occurs if either the previous state is an internal state or this transition is the first one, where q_j is a substate of q_i except an end state. The next state q_j is chosen according to *vertical transition probabilities* $\pi_{i,j}$. If the previous state is a production state or an end state, a horizontal transition from q_i to q_k occurs where q_k is a neighbor state of q_i. The next state q_k is selected with *horizontal transition probabilities* $a_{i,k}$. Let $a_{i,\mathrm{end}}$ denote the probability of a horizontal transition from q_i to q_i^{end}. Second, if we are in an end state q_i^{end}, we move up by a forced transition to the parent state of q_i. Lastly, if we are in a production state $q_{i'}$, we observe a symbol σ_h according to *output probabilities* $b_{i',h}$, and move up by a forced transition to the parent state of $q_{i'}$.

For an internal state q_i, let $sub(i)$ be a set of indices of substates of q_i except end states, and $fwd(i)$ that of indices of neighbor states of q_i except end states. For a production state $q_{i'}$, let $sym(i')$ be a set of indices of symbols that $q_{i'}$ outputs. To make probabilities $\pi_{i,j}$, $a_{i,k}$, and $b_{i,h}$ consistent with an HHMM, for any internal state q_i, it must hold that $\sum_{j \in sub(i)} \pi_{i,j} = a_{i,\mathrm{end}} + \sum_{k \in fwd(i)} a_{i,k} = 1$, and $b_{i,h} = 0$ for any h. For any production state $q_{i'}$, it must also hold $\pi_{i',j} = a_{i',j} = 0$ for any j, and $\sum_{h \in sym(i')} b_{i',h} = 1$.

A sentence generated by an HHMM is a sequence of symbols output in a state sequence from the initial state q_1 to q_1^{end}. Suppose an HHMM in Fig. 1 (a)

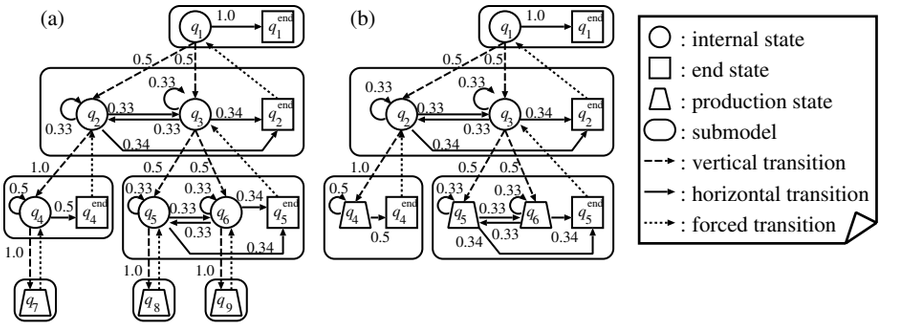


Fig. 1. Examples of HHMMs. (a) A partial-transition model. (b) A full-transition model. For any sentence o, a probability that (a) outputs o is equivalent to a probability that (b) does. q_3^{end} and q_6^{end} are identical to q_2^{end} and q_5^{end}, respectively.

outputs a sentence $\sigma_1\sigma_1\sigma_4\sigma_2\sigma_3$ such that production states q_7, q_8, and q_9 in the HHMM output σ_1, σ_2, and σ_3, respectively. Let $q_i(\sigma_j)$ stand for q_i outputting σ_j, and let $\xrightarrow{v}$, $\xrightarrow{h}$, and $\xrightarrow{f}$ represent a vertical transition, a horizontal transition, and a forced transition, respectively. One possible state sequence is the following:

$$
\begin{array}{llllllll}
q_1 & \xrightarrow{v} q_2 & \xrightarrow{v} q_4 & \xrightarrow{v} q_7(\sigma_1) & \xrightarrow{f} q_4 & \xrightarrow{h} q_4 & \xrightarrow{v} q_7(\sigma_1) & \xrightarrow{f} q_4 \\
\xrightarrow{h} q_4^{\text{end}} & \xrightarrow{f} q_2 & \xrightarrow{h} q_3 & \xrightarrow{v} q_6 & \xrightarrow{v} q_9(\sigma_3) & \xrightarrow{f} q_6 & \xrightarrow{h} q_5 & \xrightarrow{v} q_8(\sigma_2) \\
\xrightarrow{f} q_5 & \xrightarrow{h} q_6 & \xrightarrow{v} q_9(\sigma_3) & \xrightarrow{f} q_6 & \xrightarrow{h} q_5^{\text{end}} & \xrightarrow{f} q_3 & \xrightarrow{h} q_2^{\text{end}} & \xrightarrow{f} q_1 \\
\xrightarrow{h} q_1^{\text{end}}. & & & & & & &
\end{array}
$$

For horizontal transitions, there are two confusing descriptions in the original paper [6]. One says, as we have described, that horizontal transitions are allowed only from internal states.[1] The other says that horizontal transitions can occur from production states.[2] They are obviously conflicting. We name them a *partial-transition model* and a *full-transition model* for later referece, respectively. Figure 1 (a) and (b) are examples of a partial-transition model and a full-transition model, respectively. The original training algorithm called the generalized Baum-Welch algorithm [6] seems applicable to only a full-transition model. On the other hand, we adopt a partial-transition model in order to make proposed training algorithms and their derivations simple.

Lastly, for timing analysis, we evaluate the number of states in a partial-transition model M transformed from a full-transition model M'. One way to transform from M' to M is that, for each production state q_i in M', q_i is set

[1] For example, "... an HHMM is characterized by the state transition probability between the internal state and the output distribution vector of the production states." in Sect. 2 of [6].

[2] For an internal state q at depth $D-1$, a transition probability matrix $A^{q^{D-1}} = (a_{ij}^{q^{D-1}})$ is defined in Sect. 2 of [6]. This matrix contains horizontal transition probabilities $a_{ij}^{q^{D-1}} = P(q_j^D | q_i^D)$ between production states, q_i^D and q_j^D.

to an internal state, and to have a unique production state in M. Figure 1 is an example of this transformation. Let n_{int} and n'_{int} be the numbers of internal states in M and M', respectively, and n_{pro} and n'_{pro} those of production states in M and M', respectively. We then have $n_{\text{int}} = n'_{\text{int}} + n'_{\text{pro}}$ and $n_{\text{pro}} = n'_{\text{pro}}$.

2.2 The Generalized Baum-Welch Algorithm

The generalized Baum-Welch algorithm was proposed as an EM algorithm for HHMMs [6], and was analyzed to take $O(vnl^3)$ time to update parameters of an HHMM,[3] where v is the number of training sentences, n is the number of states, and l is the maximum length of training sentences.

For each training sentence $o_1, \ldots, o_{l'}$, the generalized Baum-Welch algorithm requires various probabilities $\alpha(s,t,q_i,q_k)$, $\beta(s,t,q_i,q_k)$, $\eta_{in}(s,q_i,q_k)$, $\eta_{out}(s,q_i,q_k)$, $\xi(s,q_i,q_j,q_k)$, $\gamma_{in}(s,q_i,q_k)$, $\gamma_{out}(s,q_i,q_k)$, and $\chi(s,q_i,q_k)$, where $1 \leq s \leq t \leq l'$, q_k is an internal state, and q_i and q_j are substates of q_k. We argue here with $\alpha(s,t,q_i,q_k)$, $\eta_{out}(s,q_i,q_k)$, and $\xi(s,q_i,q_j,q_k)$. $\alpha(s,t,q_i,q_k)$ is a probability that symbols $o_s \cdots o_t$ are output in any state sequences from q_k to q_i, $\eta_{out}(s,q_i,q_k)$ is one that a horizontal transition from q_i occurs before $o_{s+1}, \ldots, o_{l'}$ are output. $\xi(s,q_i,q_j,q_k)$ is one that a horizontal transition from q_i to q_j occurs between output of $o_1, \ldots, o_s$ and that of $o_{s+1}, \ldots, o_{l'}$. Due to space limitations, see the original paper [6] for further details of the other probabilities. From the viewpoint of calculating these probabilities, the generalized Baum-Welch algorithm has the following four drawbacks.

First, the definition of $\eta_{out}(l',q_i,q_k)$ in [6] is incomplete. $\eta_{out}(l',q_i,q_k)$ is recursively defined using $\eta_{out}(l',q_k,q_h)$ where q_h is a parent state of q_k. The basis, $\eta_{out}(l',q_{i'},q_1)$, is not defined properly,[4] where $q_{i'}$ is an internal state at depth 2. $\eta_{out}(l',q_i,q_k)$ is required by $\chi(s,q_g,q_i)$, and $\chi(s,q_g,q_i)$ is necessary for updating parameters $\pi_{i,g}$, where q_g is a substate of q_i. Then parameters are not updated by the generalized Baum-Welch algorithm.

Second, the generalized Baum-Welch algorithm sets a probability $\xi(l',q_{i'},q_{j'},q_1)$ of a logically impossible event in any HHMM to non-zero, where $q_{i'}$ and $q_{j'}$ are internal states at depth 2. $\xi(l',q_{i'},q_{j'},q_1)$ is a probability that a horizontal transition from $q_{i'}$ to $q_{j'}$ occurs after symbols $o_1, \ldots, o_{l'}$ are output. If a horizontal transition from $q_{i'}$ to $q_{j'}$ were possible, a sequence of vertical transitions would continue from $q_{j'}$ to some production state, and then the $(l'+1)$-th symbol of a sentence would be output. This contradicts that the length of the sentence is l'.

Third, the likelihood of a sentence o defined in the generalized Baum-Welch algorithm is not equivalent to a probability that an HHMM outputs o. The likelihood $P(o|\theta)$ is defined by $\sum_{i' \in sub(1)} \alpha(1,l',q_{i'},q_1)$. $\alpha(1,l',q_{i'},q_1)$ is a probability that an HHMM outputs $o_1, \ldots, o_{l'}$ from q_1 to $q_{i'}$, but it contains a probability that an HHMM continues to output $o_{l'+1}$.

[3] To obtain this bound, $|sub(i)|$ and $|fwd(i)|$ are implicitly assumed to be bounded by some constant for any i.

[4] $\eta_{out}(l',q_{i'},q_1)$ is recursively defined using $\eta_{out}(l',q_1,q_h)$ where q_h is a parent state of q_1, which never exists in any HHMM.

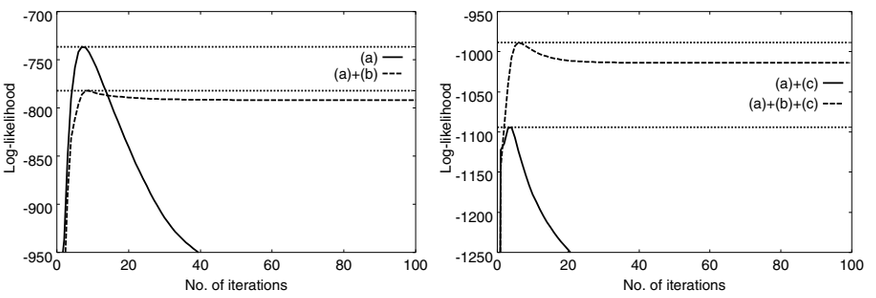


Fig. 2. Curves of the "log-likelihood" plotted by the generalized Baum-Welch algorithm with modifications: Experiments with all combinations of redefinitions (b) and (c) are demonstrated. The dotted lines show the maxima for their curves.

Therefore, to train parameters of HHMMs with the generalized Baum-Welch algorithm, we have to redefine (a) $\eta_{out}(T, q_{i'}, q_1) = a_{i',\mathrm{end}}$ to update parameters of an HHMM, (b) $\xi(l', q_{i'}, q_{j'}, q_1) = 0$, and (c) $P(o|\theta) = \sum_{i' \in sub(1)} \alpha(1, l', q_{i'}, q_1) a_{i',\mathrm{end}}$ to make the generalized Baum-Welch algorithm consistent with an HHMM.

Lastly, a proof that the generalized Baum-Welch algorithm is an EM algorithm was not contained in [6]. In addition, the algorithm sometimes showed decreases in the likelihood in our implementation. Figure 2 shows curves of the log-likelihood of 100 training sentences by the generalized Baum-Welch algorithm, given the HHMM in Fig. 1 (b) such that every production state emits any symbols. The sentences were generated randomly with the HHMM in Fig. 1 (b) such that production states q_4, q_5, and q_6 in the HHMM output σ_1, σ_2, and σ_3, respectively. This result is obviously against the fundamental property of EM algorithms in which each iteration is guaranteed to increase the log-likelihood [5]. Though we have fixed several drawbacks in the generalized Baum-Welch algorithm, it seems vain to make up our implementation into an EM algorithm without a proof that the generalized Baum-Welch algorithm is an EM algorithm. We will therefore derive a new EM algorithm for HHMMs.

3 Training Algorithms

3.1 Description with Stochastic Context-Free Grammars

Before presenting training algorithms, we show that given an HHMM M, we are able to construct a stochastic context-free grammar (SCFG) G such that a set of sentences from M is equivalent to that of sentences from G. For space limitations, we refer the reader to e.g. [4] for definitions of SCFGs.

For notational convenience, let a subsentence of q_i be symbols generated in a state sequence from q_i to either q_i^{end} or the parent state of q_i. Q_{int}, Q_{pro}, and Q_{end} are a set of internal states, that of production states, and that of internal states q_i such that a horizontal transition from q_i to q_i^{end} is available, respectively.

For a production state $q_{i'}$, recall that $sym(i')$ is a set of indices of symbols that $q_{i'}$ outputs. $A_{i'}$, a non-terminal of an SCFG, is able to generate any subsentence of $q_{i'}$ if a set of rules of an SCFG contains $A_{i'} \to \sigma_h$ for any $h \in sym(i')$. For an internal state q_i, we consider a non-terminal A_i. Any subsentence of q_i is equivalent to either w or ww' where w is a subsentence of a state q_j in a submodel of q_i, w' is a subsentence of a neighbor state q_k of q_i, and ww' is the concatenation of w and w'. Roughly speaking, A_i is able to generate any subsentence of q_i if a set of rules contains $A_i \to A_j A_k$ and $A_i \to A_j$ for any $j \in sub(i)$ and $k \in fwd(i)$.

From these observations, we can conclude that given an HHMM M, there exists an SCFG $G = (N, \Sigma, R, A_1)$ which is able to derive every sentence generated by M where $N = \{A_1, \ldots, A_n\}$,

$$R = \{A_i \to A_j A_k | q_i \in Q_{\mathrm{int}}, j \in sub(i), k \in fwd(i)\} \\ \cup \{A_i \to A_j | q_i \in Q_{\mathrm{end}}, j \in sub(i)\} \cup \{A_i \to \sigma_h | q_i \in Q_{\mathrm{pro}}, h \in sym(i)\}.$$

We then formally show that a set of sentences from G is equivalent to that from M. Let $L_M(q_i, l)$ be a set of subsentences w of q_i such that the length of w is at most l, and $L_G(A_i, l)$ be a set of symbols $w \in \Sigma^+$ such that $A_i \stackrel{*}{\Rightarrow} w$ and $|w| \leq l$. For any q_i and A_i, we set $L_M(q_i, 0) = L_G(A_i, 0) = \emptyset$. For any pair of M and G, the following holds.

Lemma 1. Let q_i be an internal state, and $l \geq 1$. Suppose $L_M(q_j, l) = L_G(A_j, l)$ and $L_M(q_k, l-1) = L_G(A_k, l-1)$ for any q_j and q_k such that q_j is a substate of q_i, and q_k is a neighbor state of q_i. It then holds that $L_M(q_i, l) = L_G(A_i, l)$.

Proposition 1. For an HHMM M and an SCFG G constructed as above, let $L_M(q_i)$ be a set of subsentences of q_i, and $L_G(A_i)$ a set of subsentences derived from A_i. For any production state or internal state q_i, $L_M(q_i) = L_G(A_i)$.

They are proved in the appendices.

For example, from the HHMM in Fig. 1 (a), we construct a set of rules R_{ex}:

$$R_{\mathrm{ex}} = \left\{ \begin{array}{lll} A_1 \to A_2 | A_3, & A_2 \to A_4 A_2 | A_4 A_3, & \\ A_3 \to A_5 A_2 | A_5 A_3 | A_6 A_2 | A_6 A_3 | A_5 | A_6, & A_4 \to A_7 A_4 | A_7, & \\ A_5 \to A_8 A_5 | A_8 A_6 | A_8, \; A_6 \to A_9 A_5 | A_9 A_6 | A_9, & & \\ A_7 \to \sigma_1, \; A_8 \to \sigma_2, \; A_9 \to \sigma_3 & & \end{array} \right\},$$

where $A_i \to \alpha_1 | \cdots | \alpha_k$ is an abbreviation for $A_i \to \alpha_1, \ldots, A_i \to \alpha_k$. With R_{ex}, the start symbol A_1 is able to derive a sentence $\sigma_1\sigma_1\sigma_4\sigma_2\sigma_3$ as follows:

$$\begin{array}{llllll} A_1 & \to A_2 & \to A_4 A_3 & \to A_7 A_4 A_3 & \to \sigma_1 A_4 A_3 & \\ \to \sigma_1 A_7 A_3 & \to \sigma_1\sigma_1 A_3 & \to \sigma_1\sigma_1 A_6 & \to \sigma_1\sigma_1 A_9 A_5 & \to \sigma_1\sigma_1\sigma_3 A_5 & \\ \to \sigma_1\sigma_1\sigma_3 A_8 A_6 & \to \sigma_1\sigma_1\sigma_3\sigma_2 A_6 & \to \sigma_1\sigma_1\sigma_3\sigma_2 A_9 & \to \sigma_1\sigma_1\sigma_3\sigma_2\sigma_3. & & \end{array}$$

Several training algorithms for SCFGs such as the Inside-Outside algorithm [1,10] and Stolcke's algorithm [13] are known. They, however, are not

usable for training an SCFG which represents an HHMM. One reason is that parameters of usual SCFGs differ from those of SCFGs describing HHMMs. In a usual SCFG, a probability of each rule is described by one parameter. On the other hand, in an SCFG describing an HHMM, we see $P(A_i \to A_j A_k|\theta) = \pi_{i,j}a_{i,k}$, $P(A_i \to A_j|\theta) = \pi_{i,j}a_{i,\mathrm{end}}$, and $P(A_i \to \sigma_j|\theta) = b_{i,h}$, where θ represents a set of parameters. That is, probabilities of several rules consist of two parameters. Hence, we need to derive new training algorithms for SCFGs describing HHMMs.

3.2 An EM Algorithm

For any SCFG describing an HHMM, the following holds similarly to usual SCFGs [9].

Proposition 2. For an SCFG which describes an HHMM, it holds that $P(O|\theta') \geq P(O|\theta)$ if any parameter $\gamma_{i,j}$ for a non-terminal A_i is updated by

$$\hat{\gamma}_{i,j} = \frac{\gamma_{i,j}}{Z_i} \sum_{u=1}^{v} \frac{1}{P(o^u|\theta)} \frac{\partial P(o^u|\theta)}{\partial \gamma_{i,j}},$$

where $P(O|\theta) = \prod_{u=1}^{v} P(o^u|\theta)$, Z_i is a normalizing constant of q_i, v is the number of training sentences, and θ' is a set of updated parameters.

Proposition 2 is proved in the appendix. From this, we are able to update parameters if we find partial derivatives $\frac{\partial P(o^u|\theta)}{\partial \gamma_{i,j}}$, the likelihood $P(o^u|\theta)$ of sentence o^u, and Z_i for any i, j, and u. For brevity, only parameters $\pi_{i,k}$ of vertical transitions will be considered here, but the derivations for the other parameters are analogous.

First, find $\frac{\partial P(o^u|\theta)}{\partial \pi_{i,j}}$. For notational convenience, we set l^u the length of the u-th sentence, $f_u(s,t,i) = P(A_1 \overset{*}{\Rightarrow} o_1^u \cdots o_{s-1}^u A_i o_{t+1}^u \cdots o_{l^u}^u|\theta)$, and $e_u(s,t,i) = P(A_i \overset{*}{\Rightarrow} o_s^u \cdots o_t^u|\theta)$.

$$\begin{aligned}
\frac{\partial P(o^u|\theta)}{\partial \pi_{i,j}} &= \frac{\partial \sum_{\tau \in \mathcal{T}(u,\pi_{i,j})} P(o^u,\tau|\theta)}{\partial \pi_{i,j}} \\
&= \sum_{s=1}^{l^u}\sum_{t=s}^{l^u} \sum_{\tau \in \mathcal{T}(u,\pi_{i,j},s,t)} P(o^u,\tau|\theta)/\pi_{i,j} \\
&= \sum_{s=1}^{l^u}\sum_{t=s}^{l^u} f_u(s,t,i)a_{i,\mathrm{end}}e_u(s,t,j) \\
&\quad + \sum_{s=1}^{l^u}\sum_{t=s+1}^{l^u} f_u(s,t,i) \sum_{k \in fwd(i)} a_{i,k} \sum_{r=s}^{t-1} e_u(s,r,j)e_u(r+1,t,k),
\end{aligned}$$

where $\mathcal{T}(u,\pi_{i,j})$ is a set of possible parse trees τ for o^u such that at least one rule with $\pi_{i,j}$ occurs in τ, i.e., τ contains $A_i \to A_j$ or $A_i \to A_j A_k$ for some k,

and $\mathcal{T}(u, \pi_{i,j}, s, t)$ a set of parse trees τ for o^u such that $A_i \stackrel{*}{\Rightarrow} o^u_s \cdots o^u_t$ and this A_i is directly derived by a rule with $\pi_{i,j}$ in τ. The second line is obtained using a formula for differentiation $\frac{(fg)(x)}{\partial x} = \frac{f(x)}{\partial x} g(x) + f(x) \frac{g(x)}{\partial x}$, and the third line is obtained by the context-free assumption. From this, $e_u(s,t,i)$ and $f_u(s,t,i)$ are required to calculate the partial derivatives.

We can find $e_u(s,t,i)$ and $f_u(s,t,i)$ recursively in a manner similar to calculation of the inner probabilities and the outer probabilities in the Inside-Outside algorithm [1,10]. We calculate the inner probability $e_u(s,t,i)$ by

$$e_u(s,t,i) = \begin{cases} b_{i,h} & s = t, \text{ where } o_s = \sigma_h, \\ 0 & s < t, \end{cases}$$

for a production state q_i, and by

$$e_u(s,t,i) = \begin{cases} \displaystyle\sum_{j \in sub(i)} \pi_{i,j} e_u(s,s,j) a_{i,\mathrm{end}} & s = t, \\ \displaystyle\sum_{r=s}^{t-1} \left(\sum_{j \in sub(i)} \pi_{i,j} e_u(s,r,j) \right) \left(\sum_{k \in fwd(i)} a_{i,k} e_u(r+1,t,k) \right) & \\ \quad + \displaystyle\sum_{j \in sub(i)} \pi_{i,j} e_u(s,t,j) a_{i,\mathrm{end}} & s < t, \end{cases}$$

for an internal state q_i. We also define the outer probability $f_u(s,t,i)$ as

$$f_u(s,t,i) = \begin{cases} 1 & s = 1,\ t = l,\ i = 1, \\ 0 & s < t,\ q_i \in Q_{\mathrm{pro}}, \\ \displaystyle\sum_{j \in bwd(i)} \sum_{k \in sub(j)} \sum_{r=1}^{s-1} f(r,t,j) \pi_{j,k}\ e_u(r, s-1, k) a_{j,i} & \\ + \displaystyle\sum_{j \in prt(i)} \sum_{k \in fwd(i)} \sum_{r=t+1}^{l^u} f_u(s,r,j)\ \pi_{j,i} e_u(t+1, r, k) a_{j,k} & \\ + \displaystyle\sum_{j \in prt(i)} f_u(s,t,j) \pi_{j,i} a_{j,\mathrm{end}} & \text{otherwise}, \end{cases}$$

where $prt(i) = \{j | i \in sub(j)\}$, and $bwd(i) = \{k | i \in fwd(k)\}$.

Second, the likelihood $P(o^u|\theta)$ is equivalent to $P(A_1 \stackrel{*}{\Rightarrow} o^u|\theta) = e_u(1, l^u, 1)$.

Lastly, it holds that $a_{i,\mathrm{end}} + \sum_{k \in fwd(i)} a_{i,k} = 1$ if we set $Z_{u,i} = \sum_{s=1}^{l^u} \sum_{t=s}^{l^u} f_u(s,t,i) e_u(s,t,i)$ and $Z_i = \sum_{u=1}^{v} Z_{u,i} / P(o^u|\theta)$ for $q_i \in Q_{\mathrm{int}}$.

Likewise, the other partial derivatives are defined as

$$\frac{\partial P(o^u|\theta)}{\partial a_{i,k}} = \sum_{j \in sub(i)} \pi_{i,j} \sum_{s=1}^{l^u} \sum_{t=s+1}^{l^u} \sum_{r=s}^{t-1} f_u(s,t,i) e_u(s,r,j) e_u(r+1,t,k),$$

$$\frac{\partial P(o^u|\theta)}{\partial a_{i,\mathrm{end}}} = \sum_{s=1}^{l^u} \sum_{t=s}^{l^u} f_u(s,t,i) \sum_{j \in sub(i)} \pi_{i,j} e_u(s,t,j),$$

$$\frac{\partial P(o^u|\theta)}{\partial b_{i,h}} = \sum_{s:o^u_s=\sigma_h} f_u(s,s,i),$$

and normalizing constants for $q_{i'} \in Q_{\mathrm{pro}}$ are defined as $Z_{i'} = \sum_{u=1}^{v} Z_{u,i'}/P(o^u|\theta)$ where $Z_{u,i'} = \sum_{s=1}^{l^u} f_u(s,s,i')e_u(s,s,i')$. By updating parameters according to the above equations, it is guaranteed to increase the likelihood of training sentences from Proposition 2.

As a summary, Fig. 3 shows a pseudo-code for this EM algorithm. It requires $O(v(n_{\mathrm{int}}l^3 + n_{\mathrm{pro}}l^2))$ time to update parameters of an HHMM, i.e., to calculate the inner probabilities, the outer probabilities, the partial derivatives, and the normalizing constants, where $n_{\mathrm{int}} = |Q_{\mathrm{int}}|$, $n_{\mathrm{pro}} = |Q_{\mathrm{pro}}|$, and $|sub(i)|$ and $|fwd(i)|$ are bounded by some constant. When a full-transition model is given, $O(vnl^3)$ time is sufficient to update parameters by this algorithm.[5] This time bound is as efficient as that of the generalized Baum-Welch algorithm, and it turns out that HHMMs are efficiently representable with SCFGs.

```
t := 0; P(O|θ^(0)) := −∞;
for all γ_{i,j} ∈ θ^(0) do
  initialize γ_{i,j} randomly s.t. Σ_j γ_{i,j} = 1 for any i;
repeat
  t := t + 1;
  for u := 1 to v do
    for ts := 0 to l^u − 1 do
      for s := 1 to l^u − ts do
        for i := n downto 1 do
          find e_u(s, s + ts, i); /* inner probabilities */
    for ts := l^u − 1 downto 0 do
      for s := 1 to l^u − ts do
        for i := 1 to n do
          find f_u(s, s + ts, i); /* outer probabilities */
    for i := 1 to n do
      find Z_{u,i}; /* normalizing constants */
    for all γ_{i,j} ∈ θ^(t−1) do find ∂P(o^u|θ)/∂γ_{i,j};
  for all γ_{i,j} ∈ θ^(t) do update γ_{i,j};
until P(O|θ^(t)) − P(O|θ^(t−1)) < ϵ;
output θ^(t);
```

Fig. 3. An EM algorithm for hierarchical hidden Markov models. $\theta^{(t)}$ stands for a set of parameters at t-th iteration.

[5] We have $v(n_{\mathrm{int}}l^3 + n_{\mathrm{pro}}l^2) = v((n'_{\mathrm{int}} + n'_{\mathrm{pro}})l^3 + n'_{\mathrm{pro}}l^2) \leq 2v(n'_{\mathrm{int}} + n'_{\mathrm{pro}})l^3 \leq 2vnl^3$, where n'_{int} and n'_{pro} are the numbers of internal states and that of production states in a full-transition model.

3.3 An Approximate Algorithm

An approximate algorithm for training HHMMs was reported to take $O(vnl^2)$ time to update parameters [6], where v is the number of training sentences, n is the number of states, and l is the maximum length of the training sentences. Unfortunately, it was neither theoretically validated nor explained in detail in [6].

We present another approximate algorithm, which takes $O(nl^3)$ time to update parameters. The idea of this algorithm is that at each iteration it selects several sentences from training sentences, and increases the log-likelihood of the selected sentences. Let $O(t) = \{o^{t,1}, \ldots, o^{t,v'}\} \subseteq O$ be a set of selected sentences at t-th iteration, where v' is bounded by some constant. Choosing sentences makes time for updating parameters independent of the number of training sentences v. In addition, this algorithm does not have to prepare all training sentences before training is started. This assures us that this algorithm makes it possible to observe sentences and train parameters simultaneously in practical tasks like robot navigation [14].

This algorithm is guaranteed to increase the likelihood of selected training sentences at each iteration, but it may decrease that of all training sentences. To avoid this drawback of the approximate algorithm in which the likelihood of all training sentences may decrease, the approximate algorithm can be combined with the EM algorithm. That is, the approximate algorithm roughly but efficiently estimates parameters in early stages, and then the EM algorithm tries to maximize the likelihood of all training sentences. This combination is called the *hybrid algorithm*.

In the hybrid algorithm, it is not clear when the EM algorithm replaces the approximate algorithm since the approximate algorithm does not require the likelihood of the whole training sentences. We then set thresholds ϵ_{hybrid} and t_{hybrid}, count the number of iterations t' at which an increase of the log-likelihood of selected sentences by the approximate algorithm is less than ϵ_{hybrid}, and heuristically switch from the approximate algorithm to the EM one when $t' \geq t_{\text{hybrid}}$.

The approximate algorithm for HHMMs is based on the smooth on-line learning algorithm for HMMs [2] since it is one of the simplest algorithms for training stochastic language models as far as we are aware. Weights $w^{\pi}_{i,j}$, $w^{a}_{i,k}$, and $w^{b}_{i,h}$ are introduced for the normalized-exponential representation [2], and parameters of an HHMM are defined as

$$\pi_{i,j} = \frac{\exp(\lambda w^{\pi}_{i,j})}{\sum_{k \in sub(i)} \exp(\lambda w^{\pi}_{i,k})},\ a_{i,k} = \frac{\exp(\lambda w^{a}_{i,k})}{\exp(\lambda w^{a}_{i,\text{end}}) + \sum_{j \in fwd(i)} \exp(\lambda w^{a}_{i,j})},$$

$$a_{i,\text{end}} = \frac{\exp(\lambda w^{a}_{i,\text{end}})}{\exp(\lambda w^{a}_{i,\text{end}}) + \sum_{j \in fwd(i)} \exp(\lambda w^{a}_{i,\text{j}})},\ b_{i,h} = \frac{\exp(\lambda w^{b}_{i,h})}{\sum_{k \in sym(i)} \exp(\lambda w^{b}_{i,k})},$$

where λ is a positive constant.

Let θ denote a set of parameters at t-th iteration, and θ' a set of updated parameters. Suppose θ' is sufficiently close to θ. We approximate the log-likelihood

of selected sentences $O(t)$ by a first order Taylor expansion around θ:

$$\log P(O(t)|\theta') \simeq \log P(O(t)|\theta) + \sum_{w^{\gamma}_{i,j}} \frac{\partial \log P(O(t)|\theta)}{\partial w^{\gamma}_{i,j}} \Delta w^{\gamma}_{i,j},$$

where $w^{\gamma}_{i,j}$ is a weight for a parameter $\gamma_{i,j}$ of an HHMM. $P(O(t)|\theta') \geq P(O(t)|\theta)$ holds if η is a small positive, and $\Delta w^{\gamma}_{i,j} = \eta \frac{\partial \log P(O(t)|\theta)}{\partial w^{\gamma}_{i,j}}$. We sketch how to calculate $\frac{\partial \log P(O(t)|\theta)}{\partial w^{\pi}_{i,j}}$ (the other partial derivatives are found in a similar way).

$$\begin{aligned}
\frac{\partial \log P(O(t)|\theta)}{\partial w^{\pi}_{i,j}} &= \sum_{k \in sub(i)} \frac{\partial \pi_{i,k}}{\partial w^{\pi}_{i,j}} \sum_{u=1}^{v'} \frac{1}{P(o^{t,u}|\theta)} \frac{\partial P(o^{t,u}|\theta)}{\partial \pi_{i,k}} \\
&= \lambda \pi_{i,j}(1 - \pi_{i,j}) \sum_{u=1}^{v'} \frac{1}{P(o^{t,u}|\theta)} \frac{\partial P(o^{t,u}|\theta)}{\partial \pi_{i,j}} \\
&\quad + \sum_{k \in sub(i) \setminus \{j\}} (-\lambda \pi_{i,j} \pi_{i,k}) \sum_{u=1}^{v'} \frac{1}{P(o^{t,u}|\theta)} \frac{\partial P(o^{t,u}|\theta)}{\partial \pi_{i,k}} \\
&= \lambda \pi_{i,j} \sum_{u=1}^{v'} \frac{1}{P(o^{t,u}|\theta)} \left(\frac{\partial P(o^{t,u}|\theta)}{\partial \pi_{i,j}} - \sum_{k \in sub(i)} \pi_{i,k} \frac{\partial P(o^{t,u}|\theta)}{\partial \pi_{i,k}} \right) \\
&= \lambda \pi_{i,j} \sum_{u=1}^{v'} \frac{1}{P(o^{t,u}|\theta)} \left(\frac{\partial P(o^{t,u}|\theta)}{\partial \pi_{i,j}} - Z_{u,i} \right),
\end{aligned}$$

where we obtain the first line by the chain rule. Likewise, for the other weights $w^{\gamma}_{i,j}$, we set

$$\Delta w^{\gamma}_{i,j} = \eta \lambda \gamma_{i,j} \sum_{u=1}^{v'} \frac{1}{P(o^{t,u}|\theta)} \left(\frac{\partial P(o^{t,u}|\theta)}{\partial \gamma_{i,j}} - Z_{u,i} \right)$$

such that $\frac{\partial P(o^{t,u}|\theta)}{\partial \gamma_{i,j}}$ and $Z_{u,i}$ are already defined in the EM algorithm. By the approximate training algorithm, it only takes $O(n_{\text{int}} l^3 + n_{\text{pro}} l^2)$ time to update parameters of an HHMM.

4 Experiment

In this section, we show an experimental result with the proposed algorithms. To compare them with the generalized Baum-Welch algorithm, we use the same data set as in the experiment in Sect. 2.2, which consists of 100 sentences generated randomly with the HHMM in Fig. 1 (b). For each algorithm, a set of rules R' $= R_{\text{ex}} \cup \{A_i \rightarrow \sigma_h | q_i \in Q_{\text{pro}}, 1 \leq h \leq 3\}$ is given.

With the EM algorithm, parameters $\pi_{i,j}$, $a_{i,k}$, and $b_{i,h}$ are initialized randomly. With the approximate algorithm and the hybrid algorithm, weights $w^{\pi}_{i,j}$,

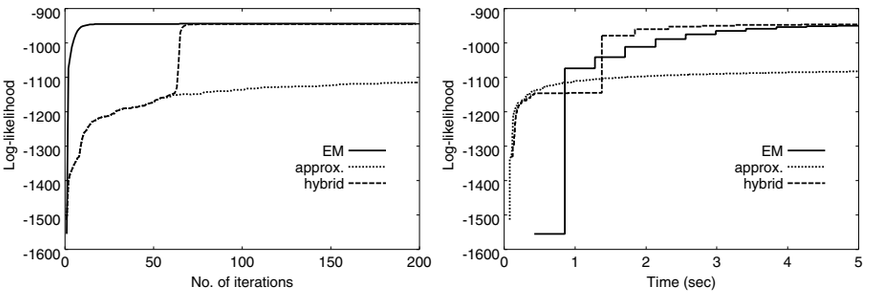


Fig. 4. Curves of the log-likelihood: (a) the log-likelihood of training sentences over iterations, (b) the log-likelihood of training sentences over time for updating parameters.

$w^a_{i,k}$, and $w^b_{i,h}$ are initialized randomly, and we put $\eta = 1.0$ ($t \leq 10$), $\eta = 10/t$ ($t > 10$), and $\lambda = 0.1$, where t is the number of iterations. For updating parameters, the EM algorithm uses all sentences, and the approximate algorithm selects sentences one by one. In the hybrid algorithm, we set $\epsilon_{\text{hybrid}} = 0.005$ and $t_{\text{hybrid}} = 20$.

For each algorithm, training was carried out 100 times in which only initial parameters (the EM algorithm) or weights (the approximate algorithm and the hybrid one) differed from each other, and ran programs on Pentium II 450 MHz with FreeBSD. Experimental results, averages of the log-likelihood of 100 sentences over iterations and those over time,[6] are shown in Fig. 4.

In (a), unlike the generalized Baum-Welch algorithm, the EM algorithm always increased the log-likelihood as we have proved. In (b), the training curve of the hybrid algorithm increased fastest. In the hybrid algorithm, the jump of the log-likelihood is caused by changing from increasing the log-likelihood to maximizing the auxiliary function $Q(\theta, \theta')$. This result is not general, but suggests that rough estimation by the approximate algorithm provide a set of plausible initial parameters for the EM algorithm, and cause time for training to be reduced.

5 Conclusion

We have derived a simplified EM algorithm and an approximate algorithm for training hierarchical hidden Markov models (HHMMs). We have also shown HHMMs are efficiently representable with stochastic context-free grammars.

We need to carry out more experiments to measure the performance of our algorithms. We are currently planning to apply these algorithms to practical domains.

[6] We did not compare the log-likelihood of the generalized Baum-Welch algorithm with those of the proposed algorithms over time since the generalized Baum-Welch algorithm does not estimate parameters correctly as we have seen in Sect 2.2.

Acknowledgments

The authors thank anonymous referees for valuable comments.

References

1. Baker, J.: Trainable grammars for speech recognition. In *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, pp. 547–550, 1979.
2. Baldi, P. and Chauvin, Y.: Smooth on-line learning algorithms for hidden Markov models. *Neural Computation*, 6 (2), pp. 305–316, 1994.
3. Bengio, Y. and Fransconi, P.: An input–output HMM architecture. *Advances in Neural Information Processing Systems* 7, pp. 427–434, 1995.
4. Charniak, E.: *Statistical language learning*, The MIT Press, 1993.
5. Dempster, A., Laird, N., and Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39, pp. 1–38, 1977.
6. Fine, S., Singer, Y., and Tishby, N.: The hierarchical hidden Markov model: analysis and applications. *Machine Learning*, 32, pp. 41–62, 1998.
7. Ghahramani, Z. and Jordan, M. I.: Factorial hidden Markov models. *Machine Learning*, 29, pp. 245–274, 1997.
8. Krogh, A., Brown, M., Mian, I. S., Hughey, R., Sjölander, K., and Haussler, D.: Hidden Markov models in computational biology: Application to protein modeling. *Journal of Molecular Biology*, 235, pp. 1501–1531, 1994.
9. Lafferty, J. D.: A derivation of the inside-outside algorithm from the EM algorithm. *IBM research report*, IBM T. J. Watson Research Center, 1993.
10. Lari, K. and Young, S. J.: The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 4, pp. 35–56, 1990.
11. McCallum, A., Freitag, D., and Pereira, F.: Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of 17th International Conference on Machine Learning*, pp. 591–598, 2000.
12. Rabiner, L. R.: A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77, pp. 257–284, 1989.
13. Stolcke, A.: An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21, pp. 165–201, 1995.
14. Theocharous, G., Rohanimanesh, K., and Mahadevan, S.: Learning and planning with hierarchical stochastic models for robot navigation. *ICML 2000 Workshop on Machine Learning of Spatial Knowledge*, 2000.

A: Proof of Lemma 1

Let q_i be an arbitrary internal state, and fix $l \geq 1$. q_j denotes a substate of q_i, and q_k a neighbor state of q_i. Suppose (a) $L_M(q_j, l) = L_G(A_j, l)$ and (b) $L_M(q_k, l-1) = L_G(A_k, l-1)$ for any q_j and any q_k.

We first show that $w_i \in L_G(A_i, l)$ if $w_i \in L_M(q_i, l)$ for any w_i. Let w_i be a subsentence of q_i such that $|w_i| \leq l$. For w_i, two cases are distinguished by horizontal transitions from q_i. One is when we move from q_i to q_i^{end} by a

horizontal transition. It then holds that $w_i = w_j$ for some $w_j \in L_M(q_j, l)$. In addition, a horizontal transition from q_i to q_i^{end} is available iff $(A_i \to A_j) \in R$. By the assumption (a), it also holds that $A_i \to A_j \stackrel{*}{\Rightarrow} w_j = w_i$.

The other is when we move from q_i to q_j by a horizontal transition. It then holds that $w_i = w_j w_k$ for some $w_j \in L_M(q_j, l)$ and $w_k \in L_M(q_k, l-1)$. In addition, a horizontal transition from q_i to q_k is available iff $(A_i \to A_j A_k) \in R$. By the assumptions (a) and (b), it also holds that $A_i \to A_j A_k \stackrel{*}{\Rightarrow} w_j w_k = w_i$. We therefore obtain that $w_i \in L_G(A_i, l)$ if $w_i \in L_M(q_i, l)$ for any w_i.

For the converse, it can be shown by similar arguments that $w_i \in L_M(q_i, l)$ if $w_i \in L_G(A_i, l)$ for any w_i. □

B: Proof of Proposition 1

The proof involves a double induction. We wish to prove $L_M(q_i, l) = L_G(A_i, l)$ by backward induction on the depth d of q_i. But in order to prove the result for d given the result for $d+1$, we must also do an induction on l.

First, let $q_{i'}$ be a state at depth D. Since $q_{i'}$ does not have any submodel, $q_{i'}$ must be a production state. For $q_{i'}$, we obtain $L_M(q_{i'}, 1) = L_G(A_{i'}, 1)$ since $h \in sym(i')$ iff $(A_{i'} \to \sigma_h) \in R$. We then assume $l \geq 2$. Now that a forced transition from $q_{i'}$ to its parent state immediately occurs after only one symbol is output, it holds that $L_M(q_{i'}, l) = L_M(q_{i'}, 1)$. On the other hand, for $A_{i'}$, R contains only rules $A_{i'} \to \alpha$ such that $\alpha \in \Sigma$. This results in that $L_G(A_{i'}, l) = L_G(A_{i'}, 1)$. From $L_M(q_{i'}, 1) = L_G(A_{i'}, 1)$, we therefore obtain $L_M(q_{i'}, l) = L_G(A_{i'}, l)$ for any state $q_{i'}$ at depth D and any l.

Second, let q_i be a state at depth $d < D$. If q_i is a production state, the discussion for a state at depth D is directly applicable to q_i. We then only have to consider that q_i is an internal state. The hypothesis for the induction on d is that $L_M(q_j, l) = L_G(A_j, l)$ for any state q_j at depth $d+1$ and any l. It holds that $L_M(q_j, 1) = L_G(A_j, 1)$ and $L_M(q_i, 0) = L_G(A_i, 0) = \emptyset$. It then follows from Lemma 1 that $L_M(q_i, 1) = L_G(A_i, 1)$. Supoose $l \geq 2$. The hypothesis for the induction on l is that $L_M(q_i, l-1) = L_G(A_i, l-1)$ for any state q_i at depth d. From $L_M(q_j, l) = L_G(A_j, l)$ and Lemma 1, it turns out that $L_M(q_i, l) = L_G(A_i, l)$.

We therefore conclude that $L_M(q_i, l) = L_G(A_i, l)$ for any q_i and any l. □

C: Proof of Proposition 2

This proof is based on that of the Inside-Outside algorithm in [9]. Let $\pi_{i,j}$, $a_{i,k}$, and $b_{i,h}$ be the current parameters. For space limitation, we consider only $\hat{\pi}_{i,j}$ as a new parameter. For notational convenience, θ and θ' denote the sets of the current parameters and the new parameters, respectively. Let τ be an arbitrary parse tree for a sentence o^u, and $C(A \to \alpha; \tau, o^u)$ be the number of times that a rule $A \to \alpha$ $(\alpha \in (N \cup T)^*)$ occurs in τ. Then

$$\log P(\tau, o^u|\theta)$$

$$
\begin{aligned}
&= \sum_{(A\to\alpha)\in R} C(A\to\alpha;\tau,o^u)\log P(A\to\alpha|\theta)\\
&= \sum_{i=1}^{n}\Bigg[\sum_{j\in sub(i)}\left(\sum_{k\in fwd(i)} C(A_i\to A_jA_k;\tau,o^u|\theta)+C(A_i\to A_j;\tau,o^u|\theta)\right)\log\pi_{i,j}\\
&\quad+\sum_{k\in fwd(i)}\left(\sum_{j\in sub(i)} C(A_i\to A_jA_k;\tau,o^u)\right)\log a_{i,k}\\
&\quad+\sum_{j\in sub(i)} C(A_i\to A_j;\tau,o^u)\log a_{i,\mathrm{end}}+\sum_{h\in sym(i)} C(A_i\to b_{i,h};\tau,o^u)\log b_{i,h}\Bigg].
\end{aligned}
$$

On the other hand, an auxiliary function $Q(\theta,\theta')$ is defined as

$$
Q(\theta,\theta')=\sum_{u=1}^{v}\sum_{\tau} P(\tau|o^u,\theta)\log P(\tau,o^u|\theta').
$$

Since $P(O|\theta')\geq P(O|\theta)$ if $Q(\theta,\theta')\geq Q(\theta,\theta)$ [5], we will maximize $Q(\theta,\theta')$ by Lagrange multiplier to increase the likelihood. The Lagrange function $\mathcal{L}(\hat{\pi}_{i,j})$ is

$$
\mathcal{L}(\hat{\pi}_{i,j})=Q(\theta,\theta')+Z_i\left(1-\sum_{j\in sub(i)}\hat{\pi}_{i,j}\right),
$$

where Z_i is a constant for i. If $\partial\mathcal{L}(\hat{\pi}_{i,j})/\partial\hat{\pi}_{i,j}$ is set to 0 to maximize $Q(\theta,\theta')$,

$$
\hat{\pi}_{i,j}=\frac{1}{Z_i}\sum_{u=1}^{v}\sum_{\tau}\frac{P(\tau,o^u|\theta)}{P(o^u|\theta)}\left(C(A_i\to A_j;\tau,o^u)+\sum_{k\in fwd(i)} C(A_i\to A_jA_k,\tau,o^u)\right)
$$

holds. Incidentally,

$$
\pi_{i,j}\frac{\partial P(o^u|\theta)}{\partial\pi_{i,j}}=\sum_{\tau}P(\tau,o^u|\theta)\left(C(A_i\to A_j;\tau,o^u)+\sum_{k\in fwd(i)} C(A_i\to A_jA_k;\tau,o^u)\right).
$$

Hence, to set $\partial\mathcal{L}(\hat{\pi}_{i,j})/\partial\hat{\pi}_{i,j}=0$, it must hold that

$$
\hat{\pi}_{i,j}=\frac{1}{Z_i}\sum_{u=1}^{v}\frac{\pi_{i,j}}{P(o^u|\theta)}\frac{\partial P(o^u|\theta)}{\partial\pi_{i,j}}.\quad\square
$$

Discovering Repetitive Expressions and Affinities from Anthologies of Classical Japanese Poems

Koichiro Yamamoto[1], Masayuki Takeda[1,2], Ayumi Shinohara[1], Tomoko Fukuda[3], and Ichirō Nanri[3]

[1] Department of Informatics, Kyushu University 33, Fukuoka 812-8581, Japan
[2] PRESTO, Japan Science and Technology Corporation (JST)
[3] Junshin Women's Junior College, Fukuoka 815-0036, Japan
{k-yama, takeda, ayumi}@i.kyushu-u.ac.jp
{tomoko-f@muc, nanri-i@msj}.biglobe.ne.jp

Abstract. The class of pattern languages was introduced by Angluin (1980), and a lot of studies have been undertaken on it from the theoretical viewpoint of learnabilities. However, there have been few practical studies except for the one by Shinohara (1982), in which patterns are restricted so that every variable occurs at most once. In this paper, we distinguish *repetitive* variables from those occurring only once within a pattern, and focus on the number of occurrences of a repetitive-variable and the length of strings it matches, in order to model the rhetorical device based on repetition of words in classical Japanese poems. Preliminary result suggests that it will lead to characterization of individual anthology, which has never been achieved, up till now.

1 Introduction

Recently, we have tackled several problems in analyzing classical Japanese poems, Waka. In [12], we successfully discovered from Waka poems characteristic patterns, named Fushi, which are read-once patterns whose constant parts are restricted to sequences of auxiliary verbs and postpositional particles. In [10], we addressed the problem of semi-automatically finding similar poems, and discovered unheeded instances of Honkadori (poetic allusion), one important rhetorical device in Waka poems based on specific allusion to earlier famous poems. On the contrary, we in [11] succeeded to discover expression highlighting differences between two anthologies by two closely related poets (e.g., master poet and disciples). In the present paper, we focus on *repetition*.

Repetition is the basis for many poetic forms. The use of repetition can heighten the emotional impact of a piece. This device, however, has received little attentions in the case of Waka poetry. One of the main reasons might be that a Waka poem takes a form of short poem, namely, it consists only of five lines and thirty-one syllables, arranged 5-7-5-7-7, and therefore the use of repetition is often considered to waste words (letters) under this tight limitation. In fact, some poets/scholars in earlier times taught their disciples *never* to repeat a word in a Waka poem. They considered word repetition as 'disease' to be avoided. This

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 416–428, 2001.

device, however, gives a remarkable effect if skillfully used, even in Waka poetry. The following poem, composed by priest Egyō (lived in the latter half of the 10th-century), is a good example of repetition, where two words 'nawo' and 'kiku' are respectively used twice [1].

HA-SHI-NO-NA-WO/NA-WO-U-TA-TA-NE-TO/KI-KU-HI-TO-NO/
KI-KU-HA-MA-KO-TO-KA/U-TSU-TSU-NA-GA-RA-NI (EGYŌ-SHŪ #195)

Since there has been few studies on this poetic device in the long research history of Waka poetry, it is necessary to develop a method of automatically extracting (candidates for) instances of the repetition from database. To retrieve instances of repetition like above, we consider the pattern matching problem for patterns such as $\star x \star x \star y \star y \star$, where $\star$ is the *variable-length don't care* (VLDC), a wildcard that matches any strings, and x, y are variables that match any non-empty strings.

Recall the *pattern languages* proposed by Angluin [2]. A *pattern* is a string in $\Pi = (\Sigma \cup V)^+$, where V is an infinite set $\{x_1, x_2, \dots\}$ of variables and $\Sigma \cap V = \emptyset$. For example, $ax_1bx_2x_1$ is a pattern, where $a, b \in \Sigma$. The *language* of a pattern π is the set of strings obtained by replacing variables in π by non-empty strings. For example, $L(ax_1bx_2x_1) = \{aubvu \mid u, v \in \Sigma^+\}$.

Although the membership problem is NP-complete for the class of Angluin patterns as shown in [2], it becomes polynomial-time solvable when the number of variables occurring within π is bounded by a fixed number k. Several subclasses have been investigated from the viewpoint of polynomial-time learnability. For example, the classes of *read-once patterns* (every variable occurs only at once) and *one-variable patterns* (only one variable is contained) are known to be polynomial-time learnable [2]. In the present paper, we try to study subclasses from viewpoints of pattern matching and similarity computation.

It should be mentioned that the class of *regular expressions with back referencing* [1] is considered as a superclass of the Angluin patterns. The membership for this class is also known to be NP-complete.

On the other hand, we attempted in [10] to semi-automatically discover similar poems from an accumulation of about 450,000 Waka poems in a machine-readable form. As mentioned above, one of the aims was to discover unheeded instances of Honkadori. The method is simple: Arrange all possible pairs of poems in decreasing order of their similarities, and then scholarly scrutinize a first part. The key to success in this approach is how to develop an appropriate similarity measure. Traditionally, the scheme of weighted edit distance with a weight matrix may have been used to quantify affinities between strings. This scheme, however, requires a fine tuning of quadratically many weights in a matrix with the alphabet size, by a hand-coding or a heuristic criterion. As an alternative idea, we introduced a new framework called *string resemblance systems* (SRSs

[1] We inserted the hyphens '-' between syllables, each of which was written as one Kana character although romanized here. One can see that every syllable consists of either a single vowel or a consonant and a vowel. Thus there can be no consonantal clusters and every syllable ends in one of the five vowels *a, i, u, e, o*.

for short) [10]. In this framework, similarity of two strings is evaluated via a pattern that matches both of them, with the support by an appropriate function that associates the quantity of resemblance candidate patterns. This scheme bridges a gap between optimal pattern discovery (see, e.g., [5]) and similarity computation.

An SRS is specified by (1) a *pattern set* to which common patterns belong, and (2) a *pattern score function* that maps each pattern in the set to the quantity of resemblance. For example, if we choose the set of patterns with VLDCs and define the score of a pattern to be the number of symbols in it, then the obtained measure is the length of the longest common subsequence (LCS) of two strings. In fact, the strings `acdeba` and `abdac` have a common pattern `a⋆d⋆a⋆` which contains three symbols.

With this framework one can easily design and modify his/her measures. In fact we designed some measures as combinations of pattern set and pattern score function along with the framework, and reported successful results in discovering unnoticed instances of Honkadori [10]. The discovered affinities raised an interesting issue for Waka studies, and we could give a convincing conclusion to it:

1. We have proved that one of the most important poems by Fujiwara-no-Kanesuke, one of the renowned thirty-six poets, was in fact based on a model poem found in Kokin-Shū. The same poem had been interpreted just to show "frank utterance of parents' care for their child." Our study revealed the poet's techniques in composition half hidden by the heart-warming feature of the poem by extracting the same structure between the two poems[2].
2. We have compared Tametada-Shū, the mysterious anthology unidentified in Japanese literary history, with a number of private anthologies edited after the middle of the Kamakura period (the 13th-century) using the same method, and found that there are about 10 pairs of similar poems between Tametada-Shū and Sōkon-Shū, an anthology by Shōtetsu. The result suggests that the mysterious anthology was edited by a poet in the early Muromachi period (the 15th-century). There have been surmised dispute about the editing date since one scholar suggested the middle of Kamakura period as a probable one. We have had a strong evidence about this problem.

In this paper, we focus on the class of Angluin patterns and on its subclasses, and discuss the problems of the pattern-matching, the similarity computation, and the pattern discovery. It should be emphasized that although many studies has been undertaken to the class of Angluin patterns and its subclasses, most of them has been done from the theoretical viewpoint of learnability. The only exception is due to Shinohara [9]. He mentioned practical applications, but they are limited to the subclass called the read-once patterns (referred to as regular patterns in [9]). We show in this paper the first practical application of Angluin

[2] *Asahi*, one of Japan's leading newspapers, made a front-page report of this discovery (26 May, 2001).

patterns that are not limited to the read-once patterns. As our framework quantifies similarities between strings by weighting patterns common to the strings, we modify the definition of patterns as follows:

- Substitute a gap symbol $\star$ for every variable occurring only once in a pattern.
- Associate each variable x with an integer $\mu(x)$ so that the variable x matches a string w only if the length of w is at least $\mu(x)$. (In the original setting in [2], $\mu(x) = 1$ for all variable x.)

Since we are interested only in repetitive strings in a Waka poem, there is no need to name non-repetitive strings. It suffices to use gap symbols $\star$ instead of variables for representing non-repetitive strings. Thus, the first item is rather for the sake of simplification. On the contrary, the second item is an essential augmentation by which the score of a pattern π can be sensitive to the values of $\mu(x)$ for variables x in π. In fact, we are strongly interested in the length of repeated string when analyzing repetitive expressions in Waka poems.

Fig. 1 is an instance of Honkadori we discovered in [10]. The two poems have several common expressions, such as, "na-ka-ra-he-te" and "to-shi-so-he-ni-ke-ru." One can notice that both the poems use the repetition of words. Namely, the Kokin-Shū poem and the Shin-Kokin-Shū repeat "nakara" (stem of verb "nagarafu"; name of a bridge) and "matsu" (wait; pine tree), respectively. This strengthens the affinities based on existence of common substrings.

Poem alluded to. (Kokin-Shū #826) Sakanoue-no-Korenori.

A-FU-KO-TO-WO	*Without seeing you,*
NA-KA-RA-NO-HA-SHI-NO	*I have lived on*
NA-KA-RA-HE-TE	*Adoring you ever*
KO-HI-WA-TA-RU-MA-NI	*Like the ancient bridge of Nagara*
TO-SHI-SO-HE-NI-KE-RU	*And many years have passed on.*

Allusive-variation. (Shin-Kokin-Shū #1636) Nijoin Sanuki.

NA-KA-RA-HE-TE	*Like the ancient pine tree of longevity*
NA-HO-KI-MI-KA-YO-WO	*On the mount of expectation called "Matsuyama,"*
MA-TSU-YA-MA-NO	*I have lived on*
MA-TSU-TO-SE-SHI-MA-NI	*Expecting your everlasting reign*
TO-SHI-SO-HE-NI-KE-RU	*And many years have passed on.*

Fig. 1. Discovered instance of poetic allusion.

It may be relevant to mention that this work is a multidisciplinary study between the literature and the computer science. In fact, the second author from the last is a Waka researcher and the last author is a linguist in Japanese language.

2 A Uniform Framework for String Similarity

This section briefly sketches the framework of string resemblance systems according to [10]. Gusfield [6] pointed out that in dealing with string similarity

the language of alignments is often more convenient than the language of edit operations. Our framework is a generalization of the alignment based scheme and is based on the notion of *common patterns*.

Before describing our scheme, we need to introduce some notation. The set of strings over an alphabet Σ is denoted by Σ^*. The length of a string u is denoted by $|u|$. The string of length 0 is called the *empty string*, and denoted by ε. Let $\Sigma^+ = \Sigma^* - \{\varepsilon\}$. Let us denote by $\boldsymbol{R}$ the set of real numbers. A *pattern system* is a triple of a finite alphabet Σ, a set Π of descriptions called *patterns*, and a function L that maps a pattern in Π to a subset of Σ^*. $L(\pi)$ is called the *language* of a pattern $\pi \in \Pi$. A pattern $\pi \in \Pi$ *match* a string $w \in \Sigma^*$ if w belongs to $L(\pi)$. A pattern π in Π is a *common pattern* of strings w_1 and w_2 in Σ^* if π matches both of them.

Definition 1. *A* string resemblance system *(SRS) is a 4-tuple $\langle \Sigma, \Pi, L, score \rangle$, where $\langle \Sigma, \Pi, L \rangle$ is a pattern system and score is a pattern score function that maps a pattern in Π to a real number.*

The *similarity* $\mathrm{SIM}(x, y)$ between strings x and y with respect to $\langle \Sigma, \Pi, L, score \rangle$ is defined by $\mathrm{SIM}(x, y) = \max\{score(\pi) \mid \pi \in \Pi \text{ and } x, y \in L(\pi)\}$. When the set $\{score(\pi) \mid \pi \in \Pi \text{ and } x, y \in L(\pi)\}$ is empty or the maximum does not exist, SIM(x,y) is undefined.

The above definition regards similarity computation as *optimal pattern discovery*. Our framework thus bridges a gap between similarity computation and pattern discovery. In [10], we defined the homomorphic SRSs and showed that the class of homomorphic SRSs covers most of the known similarity (dissimilarity) measures, such as, the edit distance, the weighted edit distance, the Hamming distance, the LCS measure. We also extended in [10] this class to the semi-homomorphic SRSs, and the similarity measures we developed in [8] for musical sequence comparison fall into this class.

We can handle a variety of string (dis)similarity by changing the pattern system and the pattern score function. The pattern systems appearing in the above examples are, however, restricted to homomorphic ones. Here, we shall mention SRSs with non-homomorphic pattern systems An *order-free pattern* (or *fragmentary pattern*) is a multiset $\{u_1, \ldots, u_k\}$ such that $k > 0$ and $u_1, \ldots, u_k \in \Sigma^+$, and is denoted by $\pi[u_1, \ldots, u_k]$. The language of pattern $\pi[u_1, \ldots, u_k]$ is the set of strings that contain the strings $u_1, \ldots, u_k$ without overlaps. The membership problem of the order-free patterns is NP-complete [7], and the similarity computation is NP-hard in general as shown in [7]. However, the membership problem is polynomial-time solvable when k is fixed. The class of order-free patterns plays an important role in finding similar poems from anthologies of Waka poems [10].

The pattern languages, introduced by Angluin [2], is also interesting for our framework.

Definition 2 (Angluin pattern system). *The Angluin pattern system is a pattern system $\langle \Sigma, (\Sigma \cup V)^+, L \rangle$, where V is an infinite set $\{x_1, x_2, \ldots\}$ of variables with $\Sigma \cap V = \emptyset$, and $L(\pi)$ is the set of strings $\pi \cdot \theta$ such that θ is a homomorphism from $(\Sigma \cup V)^+$ to Σ^+ such that $c \cdot \theta = c$ for every $c \in \Sigma$.*

In this paper we discuss SRSs with the Angluin pattern system.

3 Computational Complexity

Definition 3. Membership Problem for pattern system $\langle \Sigma, \Pi, L \rangle$.
Given a pattern $\pi \in \Pi$ and a string $w \in \Sigma^*$, determine whether or not $w \in L(\pi)$.

Theorem 1 ([2]). Membership problem for Angluin pattern system *is NP-complete.*

Definition 4. Similarity Computation with respect to SRS $\langle \Sigma, \Pi, L, score \rangle$. *Given two strings $w_1, w_2 \in \Sigma^*$, find a pattern $\pi \in \Pi$ with $\{w_1, w_2\} \subseteq L(\pi)$ that maximizes $score(\pi)$.*

Theorem 2. *For an SRS with Angluin pattern system,* Similarity Computation *is NP-hard in general.*

Proof. We consider the following problem, that is a decision version of a special case of Similarity Computation with $w_1 = w_2$, and show its NP-completeness. Optimal Pattern with respect to SRS $\langle \Sigma, \Pi, L, score \rangle$: Given a string $w \in \Sigma^*$ and an integer k, determine whether or not there is a pattern $\pi \in \Pi$ such that $w \in L(\pi)$ and $score(\pi) \geq k$.

We give a reduction from Membership Problem for Angluin pattern system $\langle \Sigma, \Pi, L \rangle$ to Optimal Pattern with respect to SRS with Angluin pattern system $\langle \Sigma', \Pi', L', score \rangle$ for a specific score function *score* defined as follows. Let $\Sigma' = \Sigma \cup \{\#\}$ with $\# \notin \Sigma$. We take a one-to-one mapping $\langle \cdot \rangle$ from $\Pi' = (\Sigma \cup V)^+$ to Σ^* that is log-space computable with respect to $|\pi|$. We define the score function $score : \Pi' \to \boldsymbol{R}$ by $score(\pi') = 1$ if π' is of the form $\pi' = \pi \# \langle \pi \rangle$ for some $\pi \in \Pi = (\Sigma \cup V)^+$, and $score(\pi') = 0$ otherwise.

For a given instance $\pi \in \Pi$ and $w \in \Sigma^*$ of Membership Problem for Angluin pattern system, let us consider $w' = w \# \langle \pi \rangle$ and $k = 1$ as an input to Optimal Pattern. Then we can see that there is a pattern $\pi' \in \Pi'$ with $w' \in L(\pi')$ and $score(\pi') = 1$ if and only if $w \in L(\pi)$, since $w' \in L(\pi')$ if and only if $\pi' = \pi \# \langle \pi \rangle$ and $w \in L(\pi)$. This completes the proof. □

4 Practical Aspects

Recall that similarities between strings are quantified by weighting patterns common to them in our framework. For a finer weighting, we augment the descriptive power of Angluin patterns by putting a restriction on the length of a string matched by each variable. Namely, we associate each variable x with an integer $\mu(x)$ such that the variable x matches a string w only if $\mu(x) \leq |w|$. For example, suppose that $\pi_1 = z_1 x z_2 x z_3$ and $\pi_2 = z_1 y z_2 y z_3$, where $\mu(x) = 2$, $\mu(y) = 3$, and $\mu(z_1) = \mu(z_2) = \mu(z_3) = 0$. Then, π_1 is common to the strings *bcaaabbaac* and *acabbaabbbb*, but π_2 is not. This enables us to define a score function so that it is sensitive to the lengths of strings substituted for variables.

On the other hand, as we have seen in the last section, similarity computation as well as membership problem is intractable in general for Angluin pattern system. From a practical point of view, it is valuable to consider subclasses of the pattern system that are tractable.

Let $occ_x(\pi)$ denote the number of occurrences of a variable x within a pattern $\pi \in (\Sigma \cup V)^+$. For example, $occ_x(abxcyxbz) = 2$. A variable x is said to be *repetitive* w.r.t. π if $occ_x(\pi) > 1$. A pattern π is said to be *read-once* if π contains no repetitive variables. Historically, read-once patterns are called *regular patterns* because the induced languages are regular [9]. The membership problem of the read-once patterns is solvable in linear time. A *k-repetitive-variable pattern* is a pattern that has at most k repetitive-variables. It is not difficult to see that:

Theorem 3. *The membership problem of the k-repetitive-variable patterns can be solved in $O(n^{2k+1})$ time for input of size n.*

That is, non-repetitive variables do not matter. Moreover, we are interested only in repeated strings in text strings. For these reasons, we substitute $\star$ for each of the non-repetitive variables in a pattern. Patterns are then strings over $(\Sigma \cup V \cup \{\star\})$, in which every variable is repetitive. For example the above pattern $abxcyxbz$ is written as $abxc \star xb\star$.

Despite the polynomial-time computability, the membership problem of the k-repetitive-variable patterns requires much time to solve. The similarity computation is therefore very slow in practice. For this reason, we in this paper restrict ourselves to the case of $k = 1$, namely, the one-repetitive-variable patterns. In order to efficiently solve the membership problem and similarity computation for this class, we utilize a kind of filtering technique. For example, when the pattern $a \star xxb \star cx$ matches a string w, then the candidate strings for substituting for x must occur at least three times in w *without overlaps*. We obtain such substring statistics on a given string w by exploiting such data structures as the *minimal augmented suffix trees* developed by Apostolico and Preparata [3,4].

Suffix tree [6] for a string w is a tree structure that represents all suffices of w as paths from the root to leaves, so that every node except leaves have at least two children. Suffix trees are useful for the task of various string processing [6]. Each node v corresponds to a substring $\tilde{v}$ of w. For each internal node v, we associate the number of leaves of the subtree rooted at v. It corresponds to the number of (possibly overlapped) occurrences $\tilde{v}$ in w to the node (see Fig. 2 (a)).

Minimal augmented suffix tree is an augmented version of the suffix tree, where additional nodes are introduced to count non-overlapping occurrences. (see Fig. 2 (b)).

5 Application to Waka Data

In this section, we present and discuss the results of our experiments carried out on *the Eight Imperial Anthologies*, the first eight of the imperial anthologies compiled by emperor commands, listed in Table 1.

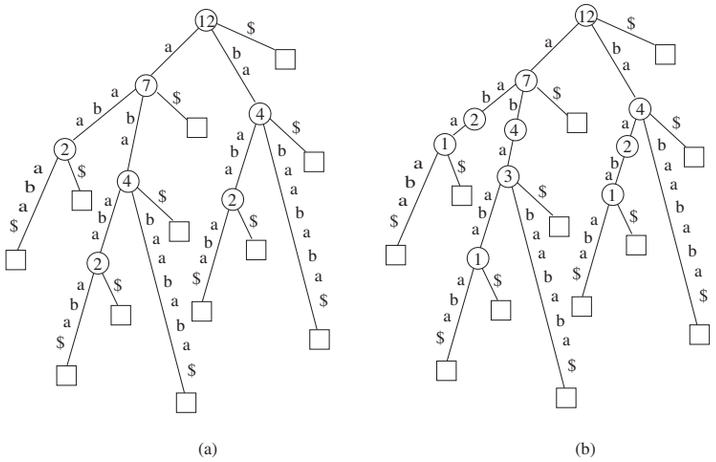


Fig. 2. (a)Suffix tree and (b)minimal augmented suffix tree for string *ababaababa*\$. The number associated to each internal node denotes the number occurrences of the string in the string, where occurrence means *possibly overlapped* occurrence in (a) and *non-overlapped* occurrence in (b). For example, the string *aba* occurs four times in the string *ababaababa*, but it appears only three times without overlapping.

Table 1. Eight Imperial Anthologies.

no.	anthology	compilation	# poems
I	Kokin-Shū	905	1,111
II	Gosen-Shū	955–958	1,425
III	Shūi-Shū	1005–1006	1,360
IV	Go-Shūi-Shū	1087	1,229
V	Kinyō-Shū	1127	717
VI	Shika-Shū	1151	420
VII	Senzai-Shū	1188	1,290
VIII	Shin-Kokin-Shū	1216	2,005

5.1 Similarity Computation

For a success in discovery, we want to put an appropriate restriction on the pattern system and on the pattern score function by using some domain knowledge. However, there are few studies on repetition of words in Waka poems as stated before, and therefore we do not in advance know what kind of restriction is effective.

We take a stepwise-refinement approach, namely, we start with very simple pattern system and score function, and then improve them based on analysis of obtained results. Here we restrict ourselves to one-repetitive-variable patterns. Moreover, we use a simple pattern score function that is not sensitive to characters or VLDCs in the patterns. Namely, the score of $a\star xxb\star cx$ is identical to that of $\star x\star x\star x\star$, for example. Despite this simplification, we wish to pay attention to

how long the strings that match variable x are. Thus, a one-repetitive-variable pattern π is essentially expressed as two integers: $occ_x(\pi)$ and $\mu(x)$. We assume that the score function is non-decreasing with respect to $occ_x(\pi)$ and to $\mu(x)$.

We compared the anthology Kokin-Shū with two anthologies Gosen-Shū and Shin-Kokin-Shū. The score function we used is defined by $score(\pi) = occ_x(\pi) \cdot \mu(x)$. The frequency distributions are shown in Table 2. From the table, there seem relatively higher similarities between Kokin-Shū and Gosen-Shū, compared with Kokin-Shū and Shin-Kokin-Shū. We examined a first part of a list of poem pairs arranged in the decreasing order of similarity value. However, we had impressions that most of pairs with high similarity value are dissimilar, probably because the pattern system we used is too simple to quantify the affinities concerning repetition techniques. See the poems shown in Fig. 3. All the poems are matched by the pattern $\star x \star x \star$ with $\mu(x) = 4$. The first three poems are similar each other, while the other pairs are dissimilar. It seems that information about the locations at which a string occurs repeatedly is important.

Table 2. Frequency distribution on similarity values in comparison of Kokin-Shū with Gosen-Shū and Shin-Kokin-Shū. Note that similarity values cannot be 1, 2, 3, 5, 7 because of the definition of the pattern score function. The frequencies for any similarity values not present here are all 0.

	0	4	6	8	10
Gosen-Shū	1,390,030	178,331	1,944	37	8
Shin-Kokin-Shū	1,962,550	244,776	2,173	11	0

KA-SU-KA-NO-HA/KE-FU-HA-NA-YA-KI-SO/WA-KA-KU-SA-NO/
TSU-MA-MO-KO-MO-RE-RI/WA-RE-MO-KO-MO-RE-RI/ (KOKIN-SHŪ #17)

TO-SHI-NO-U-CHI-NI/HA-RU-HA-KI-NI-KE-RI/HI-TO-TO-SE-WO/
KO-SO-TO-YA-I-HA-MU/KO-TO-SHI-TO-YA-I-HA-MU/ (KOKIN-SHŪ #1)

HI-RU-NA-RE-YA/MI-SO-MA-KA-HE-TSU-RU/TSU-KI-KA-KE-WO/
KE-FU-TO-YA-I-HA-MU/KI-NO-FU-TO-YA-I-HA-MU/ (GOSEN-SHŪ #1100)

HA-RU-KA-SU-MI/TA-TE-RU-YA-I-TSU-KO/MI-YO-SHI-NO-NO/
YO-SHI-NO-NO-YA-MA-NI/YU-KI-HA-FU-RI-TSU-TSU/ (KOKIN-SHŪ #3)

TSU-RA-KA-RA-HA/O-NA-SHI-KO-KO-RO-NI/TSU-RA-KA-RA-M/
TSU-RE-NA-KI-HI-TO-WO/KO-HI-M-TO-MO-SE-SU/ (GOSEN-SHŪ #592)

Fig. 3. Poems that are matched by the same pattern $\star x \star x \star$ with $\mu(x) = 4$. All pairs have a unique similarity value. The first three poems can be considered to 'share' the same poetic device and are closely similar, while some pairs are dissimilar.

Moreover, we observed that there are a lot of meaningless repetitions of strings, especially when $\mu(x)$ is relatively small, say, $\mu(x) = 2$. It seems better to restrict ourselves to repetition of strings occurring at the beginning or the end of a line in order to remove such repetitions.

We assume the lines of a poem are parenthesized by $[,]$. Then, the pattern $[\star][x\star][x\star][\star][\star]$, for example, matches any poem whose second and third lines begin with a same string. We want to use the set of such patterns as the pattern set, but the number of such patterns is $3^5 = 243$, which makes the similarity computation impractical. However, by using the Minimal Augmented Suffix Trees, we can filter out a wasteful computation and perform the computation in reasonable time. The results are shown in Table 3. By examining a first part, we confirmed that this time pairs with a high similarity value are closely similar.

Table 3. Improved results. Frequency distribution on similarity values in comparison of Kokin-Shū with Gosen-Shū and Shin-Kokin-Shū. Note that similarity values cannot be 1, 2, 3, 5, 7 because of the definition of the pattern score function. The frequencies for any similarity values not present here are all 0.

	0	4	6	8	10
Gosen-Shū	1,569,925	407	14	1	3
Shin-Kokin-Shū	2,208,888	583	39	0	0

5.2 Characterization of Anthologies

Table 4 shows the most 30 patterns occurring in Kokin-Shū. The table illustrates variations of word repetition techniques.

Table 4. Most frequent 30 patterns in Kokin-Shū.

freq.	pattern	freq.	pattern	freq.	pattern
11	$[\star][\star][x\star][x\star][\star]$	3	$[\star x][\star][\star x][\star][\star]$	1	$[x\star][\star][x\star][\star][\star x]$
10	$[x\star][x\star][\star][\star][\star]$	3	$[\star][x\star][\star][\star][x\star]$	1	$[x\star][\star][\star][\star][\star x]$
10	$[\star][x\star][x\star][\star][\star]$	3	$[\star][\star x][\star][\star x][\star]$	1	$[\star x][\star][x\star][\star][\star]$
7	$[x\star][\star][\star][\star][x\star]$	3	$[\star][\star][x\star][\star][x\star]$	1	$[\star x][\star][\star][\star][\star x]$
5	$[\star][\star x][\star][\star][\star x]$	3	$[\star][\star][\star][\star x][\star x]$	1	$[\star][x\star][\star x][\star][\star]$
5	$[\star][\star][\star x][x\star][\star]$	2	$[x\star][\star][x\star][\star][\star]$	1	$[\star][x\star][\star][x\star][\star]$
5	$[\star][\star][\star][x\star][x\star]$	2	$[\star x][\star][\star][\star x][\star]$	1	$[\star][\star x][x\star][\star][\star]$
4	$[x\star][\star][\star][x\star][\star]$	2	$[\star x][\star][\star][\star][x\star]$	1	$[\star][\star][x\star][\star x][\star]$
4	$[\star x][x\star][\star][\star][\star]$	2	$[\star][\star x][\star x][\star][\star]$	1	$[\star][\star][x\star][\star][\star x]$
4	$[\star][\star][\star x][\star x][\star]$	1	$[x\star][x\star][\star][\star][x\star]$	0	$[x\star][x\star][x\star][x\star][x\star]$

For every pattern of the above mentioned form, we collected the poems that are matched by it from the first eight imperial anthologies shown in Table 1. The results are summarized in Table 5. The first four anthologies have a

Table 5. Characterization of anthologies. I, II, III, IV, V, VI, VII, VIII represent Kokin-Shū, Gosen-Shū, Shūi-Shū, Go-Shūi-Shū, Kinyō-Shū, Shika-Shū, Senzai-Shū, Shin-Kokin-Shū, respectively,

$(occ_x(\pi), \mu(x))$	I	II	III	IV	V	VI	VII	VIII
$(2, 2)$	96	104	118	108	24	22	77	112
$(2, 3)$	23	20	28	31	5	9	17	19
$(2, 4)$	10	7	13	5	4	5	3	1
$(2, 5)$	5	5	10	3	2	2	1	0
$(3, 2)$	2	11	2	3	0	1	1	0
$(3, 3)$	0	0	0	2	0	1	0	0
$(3, 4)$	0	0	0	0	0	0	0	0
$(3, 5)$	0	0	0	0	0	0	0	0
$(4, 2)$	0	5	0	0	0	0	0	0
$(4, 3)$	0	0	0	0	0	0	0	0
$(4, 4)$	0	0	0	0	0	0	0	0
$(4, 5)$	0	0	0	0	0	0	0	0
$(5, 2)$	0	1	0	0	0	0	0	0
$(5, 3)$	0	0	0	0	0	0	0	0
$(5, 4)$	0	0	0	0	0	0	0	0
$(5, 5)$	0	0	0	0	0	0	0	0

considerable amount of poems that use repetition of words, even for a large value of $\mu(x)$. This is contrasted with Shin-Kokin-Shū where limited to a small value of $\mu(x)$. This might be a reflection of the editor's preferences or of literary trend. Anyway, pursuing the reason for such differences will provide clues for further investigation on literary trend or the editors' personalities.

6 Concluding Remarks

The Angluin pattern language has been studied mainly from theoretical viewpoints. There are no practical applications except for those limited to the read-once patterns. This paper presented the first practical application of the Angluin pattern languages that are not limited to read-once patterns. We hope that pattern matching and similarity computation for the patterns discussed in this paper possibly lead to discovering overlooked aspects of individual poets.

We distinguished repetitive variables (i.e., occurring more than once in a pattern) from non-repetitive variables, and associated each variable x with an integer $\mu(x)$ as the lower bound to the length of strings the variable x matches. This enables us to give a pattern score depending upon the lengths of strings substituted for variables. For one-repetitive-variable pattern, we presented a way

of speed-up of pattern matching, which uses substring statistics from minimal augmented suffix tree of a given string as a filter that excludes patterns which cannot match it. Preliminary experiment showed this idea successfully speeds up the pattern matching against many patterns repeatedly.

In this paper, we restricted ourselves to one-repetitive-variable patterns and to repetition of words which occur at the beginning or the end of lines of Waka poem. The restriction played an important role but we want to consider a slightly more complex patterns. For example, the following two poems are matched by the pattern $[\star][\star][x\star][xx\star][\star]$.

[SHI-RA-YU-KI-NO][YA-HE-FU-RI-SHI-KE-RU][KA-HE-RU-YA-MA]
[KA-HE-RU-KA-HE-RU-MO][O-I-NI-KE-RU-KA-NA] (KOKIN-SHŪ #902)

[A-FU-KO-TO-HA][MA-HA-RA-NI-A-ME-RU][I-YO-SU-TA-RE]
[I-YO-I-YO-WA-RE-WO][WA-HI-SA-SU-RU-KA-NA] (SHIKA-SHŪ #244)

Moreover, the next poem is matched by the pattern $[x\star][y\star][x*][x*][y*]$ that contains two-repetitive-variables.

[WA-SU-RE-SHI-TO][I-HI-TSU-RU-NA-KA-HA][WA-SU-RE-KE-RI]
[WA-SU-RE-MU-TO-KO-SO][I-FU-HE-KA-RI-KE-RE] (GO-SHŪI-SHŪ #886)

To deal with more general patterns like these ones will be future work.

References

1. A. V. Aho. *Handbook of Theoretical Computer Science*, volume A, Algorithm and Complexity, chapter 5, pages 255–295. Elsevier, Amsterdam, 1990.
2. D. Angluin. Finding patterns common to a set of strings. *J. Comput. Sys. Sci.*, 21:46–62, 1980.
3. A. Apostolico and F. Preparata. Structural properties of the string statistics problem. *J. Comput. & Syst. Sci.*, 31(3):394–411, 1985.
4. A. Apostolico and F. Preparata. Data structures and algorithms for the string statistics problem. *Algorithmica*, 15(5):481–494, 1996.
5. H. Arimura. Text data mining with optimized pattern discovery. In *Proc. 17th Workshop on Machine Intelligence*, Cambridge, July 2000.
6. D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press, New York, 1997.
7. H. Hori, S. Shimozono, M. Takeda, and A. Shinohara. Fragmentary pattern matching: Complexity, algorithms and applications for analyzing classic literary works. In *Proc. 12th Annual International Symposium on Algorithms and Computation (ISAAC'01)*, 2001. To appear.
8. T. Kadota, M. Hirao, A. Ishino, M. Takeda, A. Shinohara, and F. Matsuo. Musical sequence comparison for melodic and rhythmic similarities. In *Proc. 8th International Symposium on String Processing and Information Retrieval (SPIRE2001).* IEEE Computer Society, 2001. To appear.
9. T. Shinohara. Polynomial-time inference of pattern languages and its applications. In *Proc. 7th IBM Symp. Math. Found. Comp. Sci.*, pages 191–209, 1982.

10. M. Takeda, T. Fukuda, I. Nanri, M. Yamasaki, and K. Tamari. Discovering instances of poetic allusion from anthologies of classical Japanese poems. *Theor. Comput. Sci.* To appear.
11. M. Takeda, T. Matsumoto, T. Fukuda, and I. Nanri. Discovering characteristic expressions from literary works. *Theor. Comput. Sci.* To appear.
12. M. Yamasaki, M. Takeda, T. Fukuda, and I. Nanri. Discovering characteristic patterns from collections of classical Japanese poems. *New Gener. Comput.*, 18(1):61–73, 2000.

Web Site Rating and Improvement Based on Hyperlink Structure

Hironori Hiraishi, Hisayoshi Kato, Naonori Ohtsuka, and Fumio Mizoguchi

Information Media Center
Science University of Tokyo
Noda, Chiba, 278-8510, Japan

Abstract. This paper describes a web site rating and improvement method that automatically suggests how to improve the web site based on a hyperlink structure. First, web site visualization using the three-dimensional hyperbolic tree shows us a map of the web site. This allows us to understand the overall web site structure and to discover where information is concentrated or is missing. In visualizing the web site, the web site rating is done from six viewpoints by analyzing all descriptions of homepages contained in the web site. The rating result is then expressed as a radar chart. Furthermore, some features of branches, which contain some homepages located in a lower layer, are extracted using a machine learning technique. If no feature is extracted, we can understand that information is not well-organized in the lower-layer of the branch. In contrast, branches that have the same features are combined by an additional hyperlink. Feature extracting of each branch in a web site automatically yields generating suggestions for improving the hyperlink structure.

1 Introduction

In general, a web site structure has a disorderly extension of hyperlinks. Such a structure does not lend itself to information retrieval in advance. In this paper, we focus on how to construct a web site for information retrieval rather than information retrieval such as by a search engine. We then propose a method for web site rating based on hyperlink structure and a method for automatically generating suggestions to improve hyperlink structures.

Yakov Nielsen discussed web site design from the viewpoint of web usability in his book [Nielsen,2000]. According to his book, a web site should be constructed based on common sense of the web from a user's viewpoint. He also discussed web site structure. For example, a structure copying the organization is not so effective; it would be better to construct a web site that categorizes related information.

To evaluate a web site hyperlink structure, we first use a three-Dimensional Hyperbolic Tree to visualize the web site's overall structure. This enables us to understand the overall web site structure and to discover where information is concentrated or lacking. In visualizing the web site, the web site is rated from six viewpoints by analyzing all description of homepages contained in the web

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 429–434, 2001.

site. The rating is then expressed as a radar chart. Furthermore, some features of branches that contain some homepages located in a lower layer are extracted using a machine learning technique. If no feature is extracted, we can understand that information in the lower layer of the branch is not well-organized. Branches that have the same feature are combined by additional hyperlinks. Feature extraction of each branch in a web site automatically generates suggestions for improving the hyperlink structure.

Thus, our web rating reads all pages and presents a web site map. The rating results are expressed as the radar chart, and features of each branch are extracted to automatically generate suggestions for improving the hyperlink structure.

2 Web Site Rating

Our web site rating begins by clarifying the overall structure of a web site. We adopt a three-Dimensional Hyperbolic Tree to make a web site map (Figure 1 left). The Hyperbolic Tree represents a web site as a tree [1] that is reflected in the Hyperbolic plane. This can show us the whole aspect of homepages contained in the web site. We can understand the scale of web site and discover where information is concentrated or lacking from the three3-Dimensional Hyperbolic Tree. All pages are down loaded to make the three-Dimensional Hyperbolic Tree and are analyzed in order to extract six quantitative features expressed as a radar chart as in Figure 1 right.

- **Scale**
  This is an evaluation of the web site scale and an index indicating how many pages the web site contains.
- **Update**
  This is an evaluation of the content freshness and an index indicating how frequently the homepages are updated.
- **Link**
  This is an evaluation of navigating in the web site and an index indicating how many internal links the web site has.
- **Portal**
  This is an evaluation of hyperlinks to related sites and an index indicating how many external links to related sites exist in the web site.
- **Media**
  This is an evaluation of homepage design and an index indicating how many media, i.e., pictures or animation, are used in the web site.
- **Structure**
  This is an evaluation about web site structure and an index that shows whether the web site has the ideal structure.

[1] The top page of the web site becomes the root of the tree and the branches are expanded in the breadth-first manner. Hyperlinks to previously found pages are ignored.

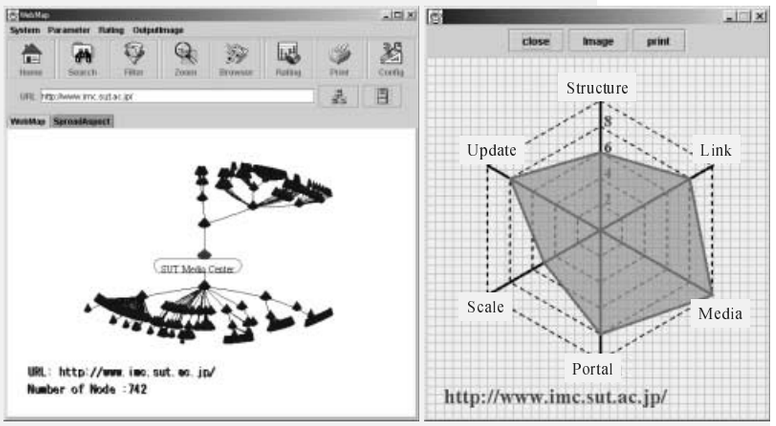


Fig. 1. Three-Dimensional Hyperbolic Tree and Rader Chart

"**Scale**" is calculated by counting the homepages in the web site. "**Update**" is calculated by getting the last modified time of the homepages. "**Portal**," "**Link**" and "**Media**" are obtained from tag information. For example, if the URL of the anchor tag is the URL of the same site, it is counted as the "**Link**"; if the URL is of a different site, it is counted as "**Portal**." "**Structure**" is evaluated by counting the pages included in each layer. The ideal quantity of pages in each layer is defined in advance, and the system evaluates whether the quantity of pages on each layer is close to the ideal quantity.

The standard structure in our method is that the quantity of pages is increased as the layer becomes deeper. The ideal web site structure is thus a pyramid in which the second layer has more pages than the first, the third layer has more than the second, etc. This is our heuristic based on the concept that there is an ideal quantity of branches from one page. In general, a web site has a directory structure that collects related information, for example, "car → maker → parts," and the number of categories tends to increase as the layer becomes deeper. Many hyperlinks in one page means that there are many branches for browsing. In this case, a possibility of the visitors deviating into the other branch becomes high, so it is not a good structure. Furthermore, a structure in which the quantity of pages becomes smaller as the layer becomes deeper may be inefficient. This occurs when many hyperlinks are in a shallow layer. In this case, in order to get target information in a deeper layer, we may have to visit many meaningless branches in shallow parts.

3 Improvement of Hyperlink Structure

We can evaluate a web site automatically from several viewpoints using our web rating method described above. If an expert evaluates a web site, we can ask the expert about improving the web site. However, it is difficult for us to understand which part should be improved and how we should improve the web site from only the evaluation value. In this section, therefore, we describe automatic generation of suggestions for improving web site structures.

Web sites should be constructed from the user's viewpoint, and it is important to categorize homepages that have same topic [Nielsen,2000]. The suggestions generated by our method point out the portions where information is distributed and where portions dealing with the same topic should be combined. That is,

"Information of this part is not well-organized.
You should combine these portions."

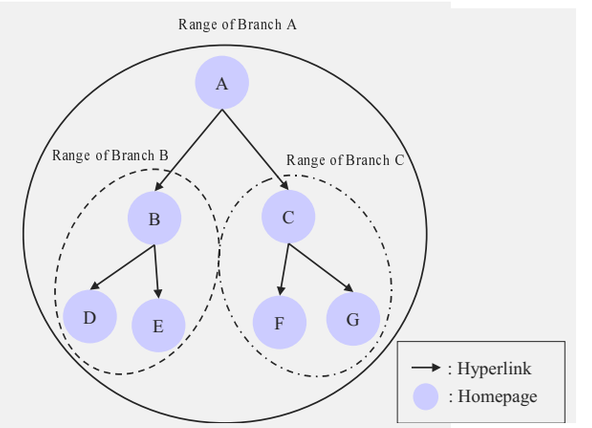


Fig. 2. Range to extract the branch feature

The most important technique for automatically providing such suggestions is to extract the "branch feature," which is a feature of homepages included in the lower layer of one node in the web site structured as a tree. Figure 2 shows the concept of the branch feature. The range of one branch is the lower part of the node, and the features of the homepages included in the branch are extracted as branch features. This clarifies what information is included in pages of the lower level. If the branch features cannot be extracted, information of pages in the branch is not well-organized. If the same branch feature is extracted from several branches, those branches can be combined as one branch.

Keywords in an HTML document can be used as the features of a homepage. They are extracted by using morphological analysis tools. Image data and tag information can also be used as features of a homepage. Rupert Parson used

keywords, URLs and hyperlink relations as features of interesting homepages for users [Parson,1998].

Figure 3 shows an example of branch features and homepage features. The end branch includes only one homepage, so the branch feature of the end branch is the same as the feature of the homepage. For example, the feature of the branch `E` is the feature of homepage `E`. Since homepage `E` has the features `a`, `b` and `e`, the branch features of `E` become `a`, `b` and `e`. Branch `B` also contains the homepages `B`, `E` and `F`. The branch features of `B` thus become `a` and `b`, which are common features among homepage `B (a,b,d)`, homepage `E (a,b,e)` and homepage `F (a,b,f)`. The branch feature of `A` is common to all homepages, so the branch feature of `A` becomes `a`.

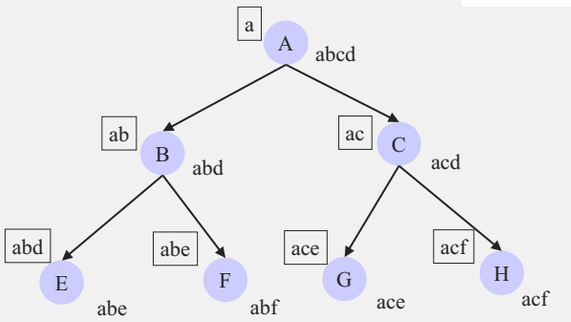


Fig. 3. Homepage and branch features

Thus, the machine learning technique can be used for extracting common features. The association rule [Agrowal,1994], which is a data mining technique, allows expression by the percentage in which this feature is included in 50% of the homepages even if the feature is not expressed in all pages. It also supports expression by the combination of features. Inductive Logic Programming (ILP) supports negative examples, so it can extract features that exist only in the branch.

4 Improving Web Site Structure

We introduce an example in which our web site rating contributed to the web site improvement. We performed free web site rating service on our web site for 10 days in November 2000 [2]. At that time, the web site shown in Figure 4 did not have a lot of pages and we could see informational deviation clearly, so it was not a good site. However, the web administrator reconstructed the web site

[2] http://imct-sev.imc.sut.ac.jp/webrating/WebRating

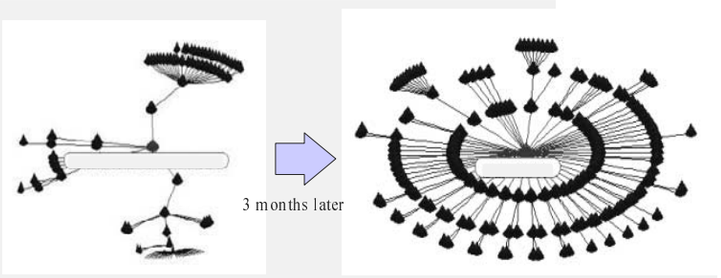


Fig. 4. Example of web site improving

after seeing this result. Three months later, the web site had changed to the orderly structure shown in Figure 4 right. Thus, our web rating can promote the improvement of a web site and provide effective web site information.

5 Conclusions

In this paper, we proposed an automatic web site rating and improvement method that suggests for improving the web site structure. Though several methods based on the hyperlink structure have been proposed to improve the efficiency of information retrieval, we have focused on the hyperlink in the web site. In so doing, we realized a system for web site rating and web site structure improvement.

Our web site rating system outputs six evaluation parameters on a radar chart. It can promote the improvement of a web site and provide effective information of the web site. In addition, our web rating of sites of a specific type enables us to discover special features.

By extracting the branch features, our system automatically suggests how to improve the web site structure. This shows us portions that are not well-organized and suggests combining branches that have similar information. It is especially effective for web sites constructed by several users.

References

[Nielsen,2000] Jakob Nielsen, *Designing Web Usability: The Practice of Simplicity*, New Riders Publishing, Indianapolis, 2000.

[Agrowal,1994] Rakesh Agrawal and Ramakrishnan Srikant, "Fast Algorithums for Mining Association Rules," *In Proc. of the 11th Conference on Very Large Databases*, 1994.

[Parson,1998] Rupert Parson and Stephen Muggleton, "An experiment with browsers that learn," *Machine Intelligence 15*, Oxford University Press, 1998.

A Practical Algorithm to Find the Best Episode Patterns

Masahiro Hirao, Shunsuke Inenaga, Ayumi Shinohara,
Masayuki Takeda, and Setsuo Arikawa

Department of Informatics, Kyushu University 33, Fukuoka 812-8581, JAPAN
{hirao, s-ine, ayumi, takeda, arikawa}@i.kyushu-u.ac.jp

Abstract. Episode pattern is a generalized concept of subsequence pattern where the length of substring containing the subsequence is bounded. Given two sets of strings, consider an optimization problem to find a best episode pattern that is common to one set but not common in the other set. The problem is known to be NP-hard. We give a practical algorithm to solve it exactly.

1 Introduction

In these days, a lot of text data or sequential data are available, and it is quite important to discover useful rules from these data. Finding a *good rule* to separate two given sets, often referred as *positive examples* and *negative examples*, is a critical task in Discovery Science as well as Machine Learning.

In [4], Hirao et al. considered *subsequence patterns* as rules. A subsequence pattern s *matches* with a string t if s can be obtained by deleting zero or more characters from t. They introduced a practical algorithm to find a best subsequence pattern that separates positive examples from negative examples, and showed some experimental results. A drawback of subsequence patterns is that they are not suitable for classifying *long* strings over *small* alphabet, since a short subsequence pattern matches with almost all long strings.

In this paper, we consider *episode patterns*, which were originally introduced by Mannila et al. [5]. An episode pattern $\langle v, k\rangle$, where v is a string and k is an integer, *matches* with a string t if v is a subsequence for some substring u of t with $|u| \leq k$. Episode pattern is a generalization of subsequence pattern since subsequence pattern v is equivalent to episode pattern $\langle v, \infty\rangle$. We give a practical solution to find a best episode pattern which separates a given set of strings from the other set of strings. We propose a practical implementation of exact search algorithm that practically avoids exhaustive search. The key idea is to introduce some heuristics to reduce the search space based on the combinatorial properties of episode patterns, and to utilize an efficient data structure that helps to determine whether an episode pattern matches with a fixed string, at the cost of preprocessing time and space requirement to construct it.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 435–440, 2001.

2 Preliminaries

Let $\mathcal{N}$ be the set of integers. Let Σ be a finite *alphabet*, and let Σ^* be the set of all *strings* over Σ. For a string w, we denote by $|w|$ the length of w. For a set $S \subseteq \Sigma^*$ of strings, we denote by $|S|$ the number of strings in S, and by $||S||$ the total length of strings in S. We say that a string v is a *prefix* (*substring*, *suffix*, resp.) of w if $w = vy$ ($w = xvy$, $w = xv$, resp.) for some strings $x, y \in \Sigma^*$. We say that a string v is a *subsequence* of a string w if v can be obtained by removing zero or more characters from w. We denote by $v \preceq_{\mathrm{str}} w$ that v is a substring of w, and by $v \preceq_{\mathrm{seq}} w$ that v is a subsequence of w. An *episode pattern* is a pair of a string v and an integer k, and we define the *episode language* $L^{\mathrm{eps}}(\langle v, k\rangle)$ by

$$L^{\mathrm{eps}}(\langle v, k\rangle) = \{w \in \Sigma^* \mid {}^{\exists}u \preceq_{\mathrm{str}} w \text{ such that } v \preceq_{\mathrm{seq}} u \text{ and } |u| \leq k\}.$$

We formulate the problem by following our previous paper [4]. Readers should refer to [4] for basic idea behind this formulation. We say that a function f from $[0, x_{\max}] \times [0, y_{\max}]$ to real numbers is *conic* if

- for any $0 \leq y \leq y_{\max}$, there exists an x_1 such that
  - $f(x, y) \geq f(x', y)$ for any $0 \leq x < x' \leq x_1$, and
  - $f(x, y) \leq f(x', y)$ for any $x_1 \leq x < x' \leq x_{\max}$.
- for any $0 \leq x \leq x_{\max}$, there exists a y_1 such that
  - $f(x, y) \geq f(x, y')$ for any $0 \leq y < y' \leq y_1$, and
  - $f(x, y) \leq f(x, y')$ for any $y_1 \leq y < y' \leq y_{\max}$.

We assume that f is conic and can be evaluated in constant time in the sequel. The following is the optimization problem to be tackled.

Definition 1 (Finding the best episode pattern according to f).
Input *Two sets $S, T \subseteq \Sigma^*$ of strings.*
Output *An episode pattern $\langle v, k\rangle$ that maximizes the value $f(x_{\langle v,k\rangle}, y_{\langle v,k\rangle})$, where $x_{\langle v,k\rangle} = |S \cap L^{eps}(\langle v, k\rangle)|$ and $y_{\langle v,k\rangle} = |T \cap L^{eps}(\langle v, k\rangle)|$.*

We remark that the problem is NP-hard, since it is a generalization of *finding the best subsequence pattern [4]*.

From the conicality of function f and the property of episode patterns, we can prove the following lemmas.

Lemma 1 ([4]). *For any $0 \leq x < x' \leq x_{\max}$ and $0 \leq y < y' \leq y_{\max}$, we have*

$$f(x, y) \leq \max\{f(x', y'), f(x', 0), f(0, y'), f(0, 0)\}.$$

Lemma 2. *For any two episode patterns $\langle v, l\rangle$ and $\langle w, k\rangle$, if $v \preceq_{seq} w$ and $l \geq k$ then $L^{eps}(\langle v, l\rangle) \supseteq L^{eps}(\langle w, k\rangle)$.*

By Lemma 1 and 2, we have the next lemma, that plays a key role in our algorithm which will be described in Section 4.

Lemma 3. *For any two episode patterns $\langle v, l\rangle$ and $\langle w, k\rangle$, if $v \preceq_{seq} w$ and $l \geq k$ then $f(x_{\langle w,k\rangle}, y_{\langle w,k\rangle}) \leq \max\{f(x_{\langle v,l\rangle}, y_{\langle v,l\rangle}), f(x_{\langle v,l\rangle}, 0), f(0, y_{\langle v,l\rangle}), f(0, 0)\}$.*

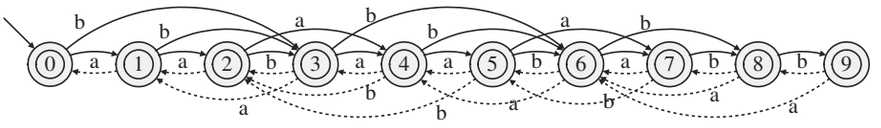


Fig. 1. *EDASG*(t), where $t = aabaababb$. Solid arrows denote the forward edges, and broken arrows denote the backward edges.

3 Episode Directed Acyclic Subsequence Graphs

We first analyze the complexity of *episode pattern matching*: given an episode pattern $\langle v, k\rangle$ and a string t, determine whether $t \in L^{\text{eps}}(\langle v, k\rangle)$ or not. This problem can be answered by filling up the edit distance table between v and t, where only insertion operation with cost one is allowed. It takes $\Theta(mn)$ time and space using a standard dynamic programming method, where $m = |v|$ and $n = |t|$.

For a fixed string, automata-based approach is useful. We use the Episode Directed Acyclic Subsequence Graph (EDASG) for string t, which was recently introduced by Troíček in [8]. A Directed Acyclic Subsequence Graph (DASG) [2] for a string t is a finite automaton that accepts all subsequences of t. An EDASG is a directed graph which combines two DASGs for t and the reversed string t^R. It contains two kinds of edges, *forward edges* corresponding to DASG(t), and *backward edges* corresponding to DASG(t^R). As an example, *EDASG*($aabaababb$) is shown in Fig. 1. When examining if an episode pattern $\langle abb, 4\rangle$ matches with t or not, we start from the initial state 0 and arrive at state 6, by traversing the forward edges spelling abb. It means that the shortest prefix of t that contains abb as a subsequences is $t[0:6] = aabaab$, where $t[i:j]$ denotes the substring $t_{i+1} \dots t_j$ of t. Moreover, the difference between the state numbers 6 and 0 corresponds to the length of matched substring $aabaab$ of t, that is, $6 - 0 = |aabaab|$. Since it exceeds the threshold 4, we move backwards spelling bba and reach state 1. It means that the shortest suffix of $t[0:6]$ that contains abb as a subsequence is $t[1:6] = abaab$. Since $6 - 1 > 4$, we have to examine other possibilities. It is not hard to see that we have only to consider the string $t[2:*]$. Thus we continue the same traversal started from state 2, that is the next state of state 1. By forward traversal spelling abb, we reach state 8, and then backward traversal spelling bba bring us to state 4. In this time, we found the matched substring $t[4:8] = abab$ which contains the subsequence abb, and the length $8 - 4 = 4$ satisfies the threshold. Therefore we report the occurrence and terminate the procedure.

With the use of *EDASG*(t), episode pattern matching can be answered quickly in practice, although the worst case behavior is still $O(mn)$. An online linear-time algorithm for constructing *EDASG*(t) for a string $t \in \Sigma^*$ was proposed in [8].

For strings $v, t \in \Sigma^*$, we define the *threshold value* θ of v for t by $\theta = min\{k \in \mathcal{N} \mid t \in L^{\text{eps}}(\langle v, k\rangle)\}$. If no such value, let $\theta = \infty$. Note that $t \notin L^{\text{eps}}(\langle v, k\rangle)$ for any $k < \theta$, and $t \in L^{\text{eps}}(\langle v, k\rangle)$ for any $\theta \leq k$. It is not difficult to see that the EDASGs are useful to compute the threshold value of v for a fixed t. We have only to repeat the above forward and backward traversal up to the end, and return the minimum length of the matched substrings.

From now on, for a set S of strings and a string v, we consider the numerical sequence $\{x_k\}_{k=0}^{\infty}$, where $x_k = |S \cap L^{\text{eps}}(\langle v, k\rangle)|$. It clearly follows from Lemma 2 that the sequence is non-decreasing. Moreover, notice that $0 \leq x_k \leq |S|$ for any k, and $x_l = x_{l+1} = x_{l+2} = \cdots$, where l is the length of the longest string in S. It implies that $\{x_k\}_{k=0}^{\infty}$ consists of at most $\min\{|S|, l\}$ distinct values. Hence we can represent $\{x_k\}_{k=0}^{\infty}$ as a list of pairs (k, x_k) such that $x_{k-1} \neq x_k$. The length of the list is bounded by $\min\{|S|, l\}$. We call this list *a compact representation of the sequence* $\{x_k\}_{k=0}^{\infty}$ (*CRS*, for short).

We now show how to compute CRS for each v and a fixed S. Observe that x_k increases only at the threshold values of v for some $t \in S$. For each string $t_i \in S$, we compute the threshold value θ_i of v for t_i, and sort these threshold values in increasing order. From these sorted values, we can construct the CRS in linear time. To be summarized, if we use the counting sort, we can compute the CRS for $v \in \Sigma^*$ in $O(|S|ml + |S|) = O(||S||m)$ time where $m = |v|$. We emphasize that the time complexity of computing the CRS of $\{x_k\}_{k=0}^{\infty}$ is the same as that of computing x_k for a single k ($0 \leq k \leq \infty$), by our method. In the next section, we use a data structure **StringSet** which supports the method to compute the CRS for any given string v.

4 Algorithm

The basic structure of the algorithm is similar to that in [4].

Fig. 2 shows our algorithm to find a best episode pattern from given two sets of strings, according to the function f. Optionally, we can specify the maximum length of episode patterns by the parameter ℓ. Here, we use a data structure **PriorityQueue** that supports the following methods.

- **bool** *empty*() : return **true** if the queue is empty.
- **void** *push*(**string** w, **double** *priority*) : push a string w into the queue with priority *priority*.
- **(string, double)** *pop*() : pop and return a pair (*string*, *priority*), where *priority* is the highest in the queue.

At line 16 marked by (*), we can simultaneously compute k' and *val* by using CRSs $\bar{x}$ and $\bar{y}$ in $O(|\bar{x}| + |\bar{y}|)$ time. By Lemma 3, we can use the value *upperBound* to prune branches in the search tree computed at line 20 marked by (**). Note that $x_{\langle v,\infty\rangle}$ and $y_{\langle v,\infty\rangle}$ can be extracted from $\bar{x}$ and $\bar{y}$ in constant time, respectively. The next theorem guarantees the completeness of the algorithm.

Theorem 1. *Let S and T be sets of strings, and ℓ be a positive integer. The algorithm FindBestEpisode(S, T, ℓ) will return an episode pattern that maxi-*

```
1  string FindBestEpisode(StringSet S, T, int ℓ)
2      string prefix, v;
3      episodePattern maxSeq; /* pair of string and int */
4      double upperBound = ∞, maxVal = −∞, val;
5      int k′;
6      CompactRepr x̄, ȳ; /* CRS */
7      PriorityQueue queue;    /* Best First Search*/
8      queue.push("", ∞);
9      while not queue.empty() do
10         (prefix, upperBound) = queue.pop();
11         if upperBound < maxVal then break;
12         foreach c ∈ Σ do
13             v = prefix+ c;    /* string concatenation */
14             x̄ = S.crs(v);
15             ȳ = T.crs(v);
16 (*)        k′ = argmax_k{f(x_⟨v,k⟩, y_⟨v,k⟩)} and val = f(x_⟨v,k′⟩, y_⟨v,k′⟩);
17             if val > maxVal then
18                 maxVal = val;
19                 maxEpisode = ⟨v, k′⟩;
20 (**)        upperBound = max{f(x_⟨v,∞⟩, y_⟨v,∞⟩), f(x_⟨v,∞⟩, 0),
                                f(0, y_⟨v,∞⟩), f(0, 0)};
21             if upperBound > maxVal and |v| < ℓ then
22                 queue.push(v, upperBound);
23     return maxEpisode;
```

Fig. 2. Algorithm *FindBestEpisode*. In our pseudocode, the **break** statement is to jump out of the closest enclosing loop.

mizes $f(x_{\langle v,k\rangle}, y_{\langle v,k\rangle})$*, with* $x_{\langle v,k\rangle} = |S \cap L^{eps}(\langle v,k\rangle)|$ *and* $y_{\langle v,k\rangle} = |T \cap L^{eps}(\langle v,k\rangle)|$*, where* v *varies any string of length at most* ℓ *and* k *varies any integer.*

5 Conclusion

We developed a practical algorithm to find the best episode pattern to separate given two sets of strings. Episode pattern is a generalization of subsequence pattern, and the search space of episode patterns is much larger than that of subsequence patterns. Nevertheless, our algorithm enabled to find the best episode pattern efficiently: the running time will *not* be much slower than that for finding subsequence patterns.

It is challenging to apply our approach to find the best *pattern* in the sense of *pattern languages* introduced by Angluin [1], where the related consistency problems are shown to be very hard [6]. Fujino et al. showed an another approach to find the best *proximity pattern* [3]. It may be interesting to combine these

approaches into one. We are now in the process of installing our algorithm into the core of the decision tree generator in the BONSAI system [7].

References

1. D. Angluin. Finding patterns common to a set of strings. *J. Comput. Syst. Sci.*, 21(1):46–62, Aug. 1980.
2. R. A. Baeza-Yates. Searching subsequences. *Theoretical Computer Science*, 78(2):363–376, Jan. 1991.
3. R. Fujino, H. Arimura, and S. Arikawa. Discovering unordered and ordered phrase association patterns for text mining. In *Proc. of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, volume 1805 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Apr. 2000.
4. M. Hirao, H. Hoshino, A. Shinohara, M. Takeda, and S. Arikawa. A practical algorithm to find the best subsequence patterns. In *Proc. of The Third International Conference on Discovery Science*, volume 1967 of *Lecture Notes in Artificial Intelligence*, pages 141–154. Springer-Verlag, Dec. 2000.
5. H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episode in sequences. In U. M. Fayyad and R. Uthurusamy, editors, *Proc. of the 1st International Conference on Knowledge Discovery and Data Mining*, pages 210–215. AAAI Press, Aug. 1995.
6. S. Miyano, A. Shinohara, and T. Shinohara. Polynomial-time learning of elementary formal systems. *New Generation Computing*, 18:217–242, 2000.
7. S. Shimozono, A. Shinohara, T. Shinohara, S. Miyano, S. Kuhara, and S. Arikawa. Knowledge acquisition from amino acid sequences by machine learning system BONSAI. *Transactions of Information Processing Society of Japan*, 35(10):2009–2018, Oct. 1994.
8. Z. Troníček. Episode matching. In *Proc. of 12th Annual Symposium on Combinatorial Pattern Matching*, Lecture Notes in Computer Science, pages 143–146. Springer-Verlag, July 2001.

Interactive Exploration of Time Series Data

Harry Hochheiser[1] and Ben Shneiderman[2]

[1] Department of Computer Science and Human-Computer Interaction Lab, University of Maryland, College Park MD 20742, +1 301 405 2725 hsh@cs.umd.edu
[2] Department of Computer Science, Human-Computer Interaction Lab, Institute for Advanced Computer Studies, and Institute for Systems Research, University of Maryland, College Park MD 20742, +1 301 405 2680 ben@cs.umd.edu

Abstract. Widespread interest in discovering features and trends in time- series has generated a need for tools that support interactive exploration.This paper introduces timeboxes: a powerful direct-manipulation metaphor for the specification of queries over time series datasets. Our TimeSearcher implementation of timeboxes supports interactive formulation and modification of queries, thus speeding the process of exploring time series data sets and guiding data mining.

1 Introduction

Interest in time series data has prompted a substantial body of work in the development of algorithmic methods for searching temporal data [1,5]. These methods would be more widely employed if the difficulty of query formulation was reduced. In order to build understanding of time series data users need tools that support data exploration via easy construction of queries and rapid feedback (100ms) [7].

Dynamic queries [2]and related information visualization techniques [4] have proven useful in meeting these goals. This paper introduces timeboxes: a dynamic query mechanism for specifying queries on temporal data sets.

2 Related Work

Data mining research has led to the development of useful techniques for analyzing time series data, including dynamic time warping [10] and Discrete Fourier Transforms (DFT) in combination with spatial queries [5]. To date, this work has paid little attention to query specification or interactive systems. One exception is Agrawal et al.'s Shape Definition Language, which specifies queries in terms of natural language descriptions of profiles [1]. Support for progressive refining of queries was addressed by Keogh and Pazanni, who suggested the use of relevance feedback for results of queries over time series data [6]. Our work with timeboxes is aimed at developing tools to address issues of user interaction with these data mining tools.

Existing time series visualizations tools generally focus on visualization and navigation, with relatively little emphasis on querying data sets. QuerySketch is an innovative query-by-example tool that uses an easily drawn sketch of a time series profile to retrieve similar profiles, with similarity defined by Euclidean distance [9]. Spotfire's Array Explorer 3 [8] supports graphically edit-able queries of temporal patterns, but the result set is generated by complex metrics in a multidimensional space.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 441–446, 2001.

3 Timeboxes: Interactive Temporal Queries

Timeboxes are rectangular query regions drawn directly on a two-dimensional display of temporal data. The extent of the timebox on the time (x) axis specifies the time period of interest, while the extent on the value (y) axis specifies a constraint on the range of values of interest in the given time period. More specifically, a timebox that goes between (x_{min}, y_{min}) and (x_{max}, y_{max}) indicates that for the time range $x_{min} \leq x \leq x_{max}$, the dynamic variable must have a value in the range $y_{min} \leq y \leq y_{max}$.

Timeboxes are created, moved, and resized using rectangle manipulation operations familiar to users of drawing and presentation software. Multiple timeboxes can be combined to specify conjunctive queries.

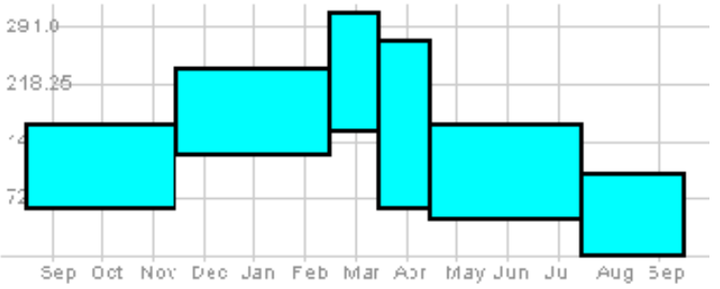


Fig. 1. Query containing multiple timeboxes

Table 1. Constraints for query shown in Fig. 1

$\forall_{\text{sep} \leq x \leq \text{nov}}\ 57 \leq y \leq 160$	$\forall_{\text{dec} \leq x \leq \text{feb}}\ 124 \leq y \leq 230$	$\forall_{x=\text{mar}}\ 154 \leq y \leq 291$
$\forall_{x=\text{apr}}\ 58 \leq y \leq 266$	$\forall_{\text{may} \leq x \leq \text{jul}}\ 46 \leq y \leq 162$	$\forall_{\text{aug} \leq x \leq \text{sep}}\ 0 \leq y \leq 101$

Fig. 1 provides an example query containing multiple timeboxes. In addition to being succinct and easy to create, the timebox version of this query provides a visual picture of the constraints that is not apparent in other notations. For example, the query in Fig. 1 is more easily interpreted than the mathematical expression of the same constraints (Table 1), which is cognitively more difficult for users to comprehend.

4 TimeSearcher

4.1 Overview

The main TimeSearcher window is shown in Fig. 2. Entities in the data set are displayed in a window in the upper left-hand corner of the application. This provides a scrollable

list that can be used to browse through the data. Complete details about the entity (details-on-demand) can be retrieved by simply clicking on the graph for the desired entity: this will cause the relevant information to be displayed in the upper right-hand window (Fig. 2).

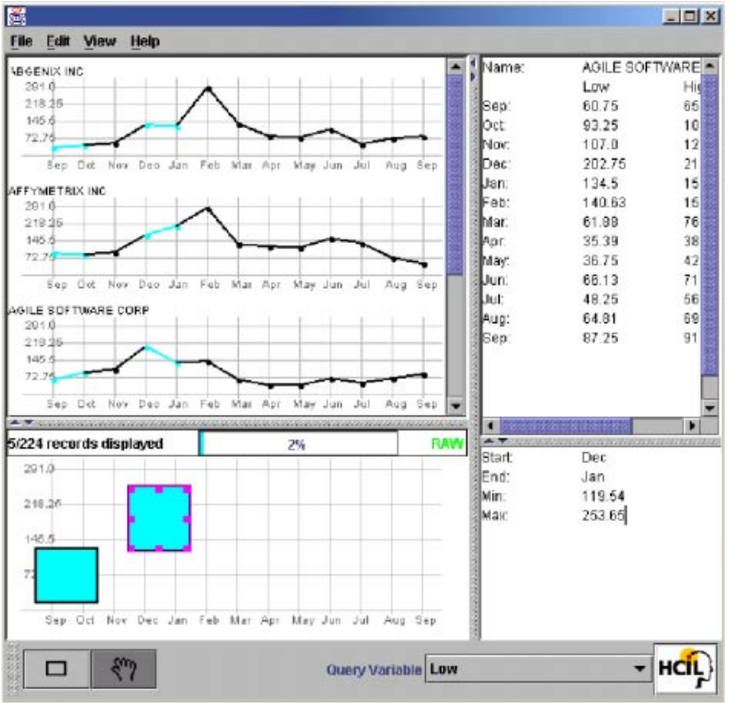


Fig. 2. TimeSearcher, displaying a query with two timeboxes and four of the five records in the result set

4.2 Query Creation and Modification

Queries are created in the query space in the bottom-left corner of the window. To specify a query, users draw a timebox in the desired location. Query processing begins as soon as users release the mouse, signifying the completion of the box. No "run" or "query" button is necessary because of the rapid update (a few hundred milliseconds). When query processing completes, the display in the top half of the application window is updated to show those entities that match the query constraints.

Rapid and dynamic update of the result set display provides prompt feedback regarding the results of the query. Once the initial query is created, query parameters can be changed by moving and resizing the timeboxes, either individually or simultaneously in groups.

4.3 Drag and Drop

Users might be interested in identifying entities that have profiles similar to a given template or example from the data set. TimeSearcher provides a drag-and-drop mechanism that can be used to identify items similar to a given example from the data set. The user can instantiate a query by dragging an item from the data display window and dropping it onto the query space. The resulting query has a separate timebox for each time point in the data set (Fig. 3). Once the query is created, the user can modify the timeboxes to modify the definition of "similar".

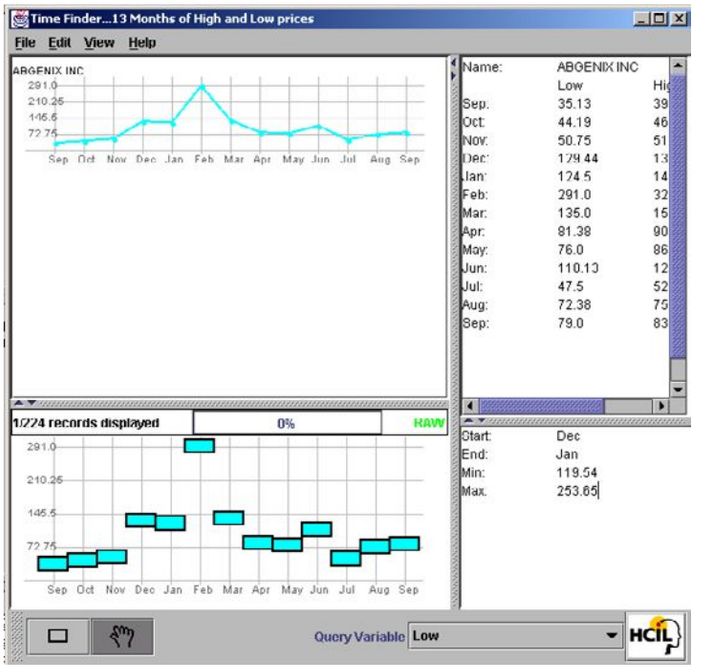


Fig. 3. Drag-and-drop query-by-example

4.4 Envelopes for Overviews

TimeSearcher uses envelopes to provide overview displays to help users make sense of large data sets [4,7]. Optionally shown in the background of the query window, the data envelope is a contour that follows the extreme values of the query attribute at each point in time, thus displaying the range of values that may be queried. When the user executes a query, the data envelope is extended by a query envelope - an overlay that outlines extreme values of the entities in the result set (Fig. 4). This display provides users with a graphic summary of the relationship between the result set and the data set as a whole.

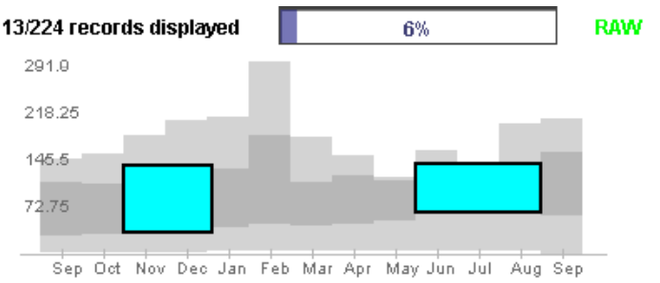


Fig. 4. Data and query envelopes for a query with two timeboxes

5 Software

TimeSearcher was implemented in Java 2, using the Swing toolkit. Drawing and scene-graph control in the data and query displays, along with functionality for moving and rescaling timeboxes, is provided by Jazz [3]. Timeboxes, graphs of each item, and query and data envelopes are implemented as Jazz widgets.

Orthogonal range trees are used to index the data, with each timebox acting as an orthogonal range query. In this model, each timebox is an orthogonal range query of width w, and an entity from the data set must have w points that fall within the query range to be included in the result set for the query.

6 Discussion and Future Work

TimeSearcher users an "overview-first" [7] approach to the exploration of time series data. The data and query envelopes, together with the linear list of graphed elements, provide the necessary overview. Each timebox is a new filter that restricts the data set resulting from the query formed by the pre-existing timeboxes. Query processing on mouse release follows a model familiar to users of modern GUIs, whereby a mouse release is treated as completion of user input.

Several extensions to the timebox model might increase the range of queries that can be expressed. Queries involving events of fixed duration occurring at any point in time, events that are separated by minimum gaps in time, disjunctions and negations, trends involving relative changes ("increase of more than 50% within a given period") and multiple time-dependent attributes might be of interest.

Further gains in efficiency might be realized by using timeboxes to specify queries to be evaluated with existing data mining algorithms such as those described by Faloutsos, et al. [5]. In this model, TimeSearcher might be used to interactively search subsets of a larger data set, in order to refine queries that might be executed against the entire data set, using the more expensive data mining algorithms.

7 Conclusions

TimeSearcher uses dynamic queries, overviews, and other information visualization techniques that have proven useful in a variety of other domains [2,4,7] to support interactive examination of time series data. Timeboxes represent an extension of the dynamic query idea to include widgets that query multiple dimensions simultaneously, as each timebox specifies constraints over two dimensions.

The incorporation of data mining algorithms into systems that support exploration and interactive knowledge discovery is the next step in making data mining more accessible to a wider range of users and problem domains. A more diverse user population will also stimulate more research, as these users generate questions and problems involving further algorithmic challenges.

The utility of timeboxes will be a function of the usability of the interface, particularly in comparison with alternative approaches. Empirical studies and heuristic evaluations are needed to clarify the benefits and drawbacks of timeboxes, while suggesting additional interface improvements.

Acknowledgments. Thanks to Martin Wattenberg for providing stock price datasets, and to Eric Baehrecke and Hyunmo Kang for valuable feedback. The first author was supported by a fellowship from America Online.

References

1. Agrawal, R., Psaila, G., Wimmers, E., and Zaït, M. Querying Shapes of Histories. In Proceedings of 21st VLDB Conference (Zurich Switzerland, September 1995), 502–514.
2. Ahlberg, C., and Shneiderman, B. Visual Information Seeking: Tight bCoupling of Dynamic Query Filters with Starfield Displays. In Proceedings of CHI '94 (Boston MA, April 1994), ACM Press, 313–317.
3. Bederson, B.B., Meyer, J., and Good, L. Jazz: an Extensible Zoomable User Interface Graphics Toolkit in Java. In Proceedings of UIST 2000 (San Diego CA, November 2000), ACM Press, 171–180.
4. Card, S.K, Mackinlay, J, D. and Shneiderman, B. Readings in Information Visualization: Using Vision to Think. Morgan-Kaufmann Publishers, San Francisco, CA, 1999.
5. Faloutsos, C., Ranganathan, M., Manolopoulos, Y. Fast Subsequence Matching in Time Series Databases. In Proceedings of SIGMOD '94 (Minneapolis MN, May 1994), ACM Press, 419–429.
6. Keogh, E.J., and Pazzani, M. J. Relevance Feedback Retrieval of Time Series Data. In Proceedings SIGIR '99 (Berkeley, CA, August 1999), ACM Press, 183–190.
7. Shneiderman, B., Designing the User Interface. Addison-Wesley, Reading, MA, 1998.
8. Spotfire. http://www.spotfire.com. (Accessed July, 2001).
9. Wattenberg, M. Sketching a Graph to Query a Time Series Database. In Proceedings of CHI 2001, Extended Abstracts (Seattle WA, April 2001), ACM Press, 381–382.
10. Yi, B.K., Jagadish, H.V., and Faloutsos, C. Efficient Retrieval of Similar Time Sequences Under Time Warping. In Proceedings of the International Conference On Data Engineering (ICDE '98), IEEE Computer Society Press, 201–208.

Clustering Rules Using Empirical Similarity of Support Sets

Shreevardhan Lele[1], Bruce Golden[1], Kimberly Ozga[2], Edward Wasil[3]

[1] University of Maryland, R.H. Smith School of Business, College Park, MD 20742, USA
{SLele, BGolden}@RHSmith.umd.edu

[2] University of Maryland, Department of Mathematics, College Park, MD 20742, USA
kaj@math.umd.edu

[3] American University, Kogod School of Business, Washington DC 20016, USA
ewasil@american.edu

Abstract. We consider the problem of pruning a given set of if-then rules, such that the support of the pruned rule set is not much less than the support of the given rule set. An empirical measure of similarity between two rules is introduced. This similarity measure is proportional to the degree of overlap between the support sets of the two rules. Using this similarity measure, we cluster the given rule set via the complete linkage algorithm. Rules within a cluster are approximate substitutes for each other and, as such, they can be replaced by a single rule, which is chosen to be the rule whose individual support value is the largest in the cluster. The pruning procedure is demonstrated on a set of rules generated from a marketing data set.

1 Introduction

If-then rules are arguably the most popular product of data mining in business. They can be generated in an unsupervised manner as association rules [1], [2] or in a supervised manner as classification rules [3]. Insights discovered from such rules are used in database marketing to effectively target products and promotions, in personalization technologies such as recommender systems to increase cross-selling and up-selling, and in designing better shelf layouts in retail outlets [4].

Several methods are available for generating association and classification rules. Regardless of the method of production, the number of rules generated is typically too large to allow for direct managerial action. Rule mining systems sift through the initial set of rules extracted and seek to retain only a subset of useful rules.

In this paper, we consider the problem of pruning a given set of rules such that there is only a minimal loss of support with respect to a given data set. Our pruning procedure uses cluster analysis to remove redundant rules

Our approach is characterized by two important features. First, unlike much of the research on rules which focuses on performance measures of *individual* rules (such as rule support, rule coverage, rule confidence, and rule simplicity), we consider a performance measure for the rule set, namely, set support. This is an important

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 447-451, 2001.

distinction because aggregating rules that are individually good does not guarantee good performance properties for the aggregated rule set.

Second, our approach employs cluster analysis in a very different fashion than previous uses of clustering in rule pruning. For example, in [5], clustering is used to combine rules based on similarities in their logical construction. On the other hand, our use of clustering is based on similarities in the support sets of the rules being clustered. If two rules point to the same class (i.e., same consequent value), we consider them to be substitutes for each other if they act on the same (or similar) sets of data records, *regardless* of the similarity or dissimilarity in their logical construction. Thus, our approach is strongly empirically driven.

Our approach can be described as a three-step procedure. The first step is to partition the initial set of rules into groups based on the predicted consequent. In the next step, we partition each such rule group into distinct clusters such that antecedents of rules within a given cluster point to similar record sets. This requires the definition of a similarity measure for rules, which we provide below. Since the record sets of their antecedents are similar and the record sets of their consequents are identical, it follows that the support sets of the various rules within a cluster are close to each other. Consequently, such rules can be viewed as substitutes for each other. In the third step, rules within each cluster are ranked on the basis of their support values and only the top rule is retained. The final rule set is obtained by aggregating the top rules from the various clusters. By construction, such a rule set will approximately span the support set of the initial set of rules, while being of a much smaller size.

2 Methodology

Let D be the data set of records on which the rules are to be applied. Let the i^{th} rule be of the form $R_i : X_i \Rightarrow Y_i$, where X_i is the antecedent and Y_i is the consequent. Let A_i denote the record set of the antecedent X_i (i.e., the subset of records in D that satisfy X_i), and let C_i denote the record set of the consequent Y_i. Let $S_i \equiv A_i \cap C_i$ denote the support set of R_i, i.e., the set of records in D where the rule R_i applies and is true. Let $\#(S)$ denote the cardinality of any set S. The support value of a rule R_i is $\#(S_i)/\#(D)$. The coverage of R_i is $\#(A_i)/\#(D)$. The confidence of R_i is $\#(S_i)/\#(A_i)$.

Consider a rule set R containing n rules $R_1, R_2, \ldots, R_n$ such that they all have a common consequent. The support set of rule set R is $S = S_1 \cup S_2 \cup \ldots \cup S_n$, and the support value of rule set R is $\#(S)/\#(D)$.

Consider two rules R_i and R_j. We define $\mathrm{sim}(i, j)$, a bounded measure of the similarity between R_i and R_j, as

$$\mathrm{sim}(i, j) \equiv \#(S_i \cap S_j) \,/\, \min\{\#(S_i), \#(S_j)\} .$$

If two rules have different consequents (i.e., different predicted classes), then their support sets will be disjoint and therefore our similarity measure will be 0. If two rules have the same consequent, then our similarity measure indicates the degree of overlap between the record sets of their antecedents. If the record sets of their antecedents are disjoint, then their similarity will be 0. At the other extreme, if the

record set of one of the antecedents is a subset of the record set of the other antecedent, then their similarity will be 1.

Given a set of rules, we first sort them on the basis of their consequents. Typically, the consequent has two possible values (buy, no-buy), although several values must be considered when predicting the purchase of multiple products. Consider the set of rules having a particular value of the consequent. For such a set, we may still have several hundred unique rules. Since these rules cannot differ in their consequent, they differ only in their antecedent. However, there may be significant redundancies when there is a large overlap in the record sets of the antecedents. Our similarity measure captures the degree of this redundancy.

For a set of n rules having a common value of the consequent, we construct an $n \times n$ similarity matrix based on our measure. Next, we use cluster analysis [6] based on this similarity matrix to partition the rule set into disjoint clusters. This can be implemented via a hierarchical agglomerative technique such as one of the various linkage algorithms. In particular, the complete linkage algorithm is known to produce compact clusters.

Selecting the number of clusters to be formed is a sensitive process. Since hierarchical clustering methods indicate the similarity level at which various clusters are merged, we can control the final number of clusters by specifying a threshold similarity level. For example, if this threshold level is set at 0.3, then the number of clusters formed is such that the similarity between any two clusters is at most 0.3.

Rules within a cluster are guaranteed to be similar to each other in the sense that the record sets satisfying their antecedents are significantly overlapping. (They also have a common consequent by construction.) As such, they can be replaced by a single rule from the cluster, or more generally, by a smaller number of rules than the cluster size.

The next step is to select a single rule (or a small number of rules) from each cluster. To accomplish this, we sort the rules within each cluster on the basis of their support measure and retain only the top few rules from each cluster.

Finally, the retained rules from each cluster are aggregated into a final rule set. Since each cluster corresponds to a distinct region in the data space (or the space of customer attributes) with respect to the antecedents, we have retained at least one rule for each populated part of the space of customer attributes. Thus, the space of customer attributes is effectively spanned by the smaller set of final retained rules.

3 Example

We demonstrate our methodology to reduce the number of classification rules obtained from a large marketing data set. The goal of all rules produced from this data set is to determine whether a given customer will re-use a particular product or not. The data set consists of 438,808 records, of which 21.63% were re-users. A training set of 60,000 records was created by random sampling. Using the popular C4.5 program [7], an initial rule set of 90 rules was extracted from the training set. All rules indicate re-use by the customer. A similarity matrix was constructed for these 90 rules using the similarity measure defined in Section 2. Complete linkage

Table 1. Performance measures of successively pruned rule sets

Size of Rule Set	Support		Coverage		Confidence	
	Training	Test	Training	Test	Training	Test
90	0.0775	0.0762	0.1586	0.1587	0.4878	0.4804
69	0.0769	0.0757	0.1571	0.1569	0.4896	0.4823
45	0.0736	0.0729	0.1471	0.1470	0.5004	0.4960
35	0.0711	0.0706	0.1403	0.1402	0.5071	0.5017
25	0.0687	0.0680	0.1337	0.1339	0.5137	0.5083
20	0.0678	0.0671	0.1309	0.1311	0.5180	0.5116
12	0.0621	0.0616	0.1155	0.1157	0.5378	0.5324

clustering was performed on this similarity matrix with a scaled threshold similarity level of 0.5. This resulted in 69 clusters, of which 57 clusters contained a single rule, while the other 12 contained more than one rule. The rules in each of the 69 clusters were sorted on the basis of their individual support values, and the rule with the maximum support was retained from each cluster.

These 69 rules were then further pruned by dropping rules with low values of support. In turn, we looked at nested rule sets containing 45, 35, 25, 20, and 12 rules. For each rule set, we computed the support, coverage, and confidence values on the training set of 60,000 records, as well as on a test set of 10,000 records randomly sampled from the original data set. These values are displayed in Table 1. The performance measures for the test set are charted in Figure 1.

We observe that the numbers for the test set show a similar pattern to those for the training set. In comparing the initial set of 90 rules with the post-clustering rule set of 69 rules, we see that the pruning of 21 rules (23% of the initial rule set) is accompanied by only a small drop in the value of support. Subsequent rows in Table 1 show that while further pruning brings the benefit of a smaller rule set, it is accompanied by progressively larger drops in support.

As expected, the coverage of the retained rule set decreases with each pruning step. However, the confidence of the retained rule set increases with each pruning step, showing that successive retained sets consist of smaller numbers of more accurate rules.

4 Conclusions

We have presented a method for rule pruning that is based on cluster analysis. Instead of looking at the logical construction of various rules to look for similarities, we look at the support sets of the rules and look for the degree of overlap. A similarity measure that quantifies this degree of overlap is defined. When the rules are clustered based on this similarity measure, rules within a cluster are relatively redundant with respect to each other since they are used to classify the same set of data records. Consequently, each cluster can be represented by a single rule without much loss of support. This results in a pruned rule set whose size is much smaller than the original rule set but without any significant decrease in support.

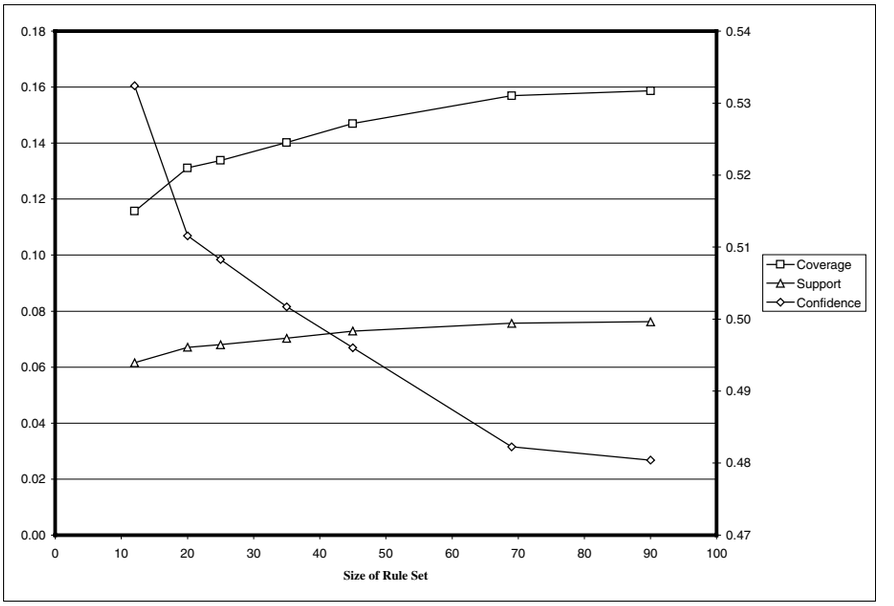


Fig. 1. Performance measures on test data set for different sizes of rule sets. Support and coverage follow scale on left, confidence follows scale on right.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast Discovery of Association Rules. In Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): Advances in Knowledge Discovery and Data Mining. AAAI Press, Menlo Park, California (1996) 307-328
2. Han, J., Kamber, M.: Data Mining. Morgan Kaufmann, San Francisco, California (2001)
3. Hand, D.: Construction and Assessment of Classification Rules. Wiley, Chichester, England (1997)
4. Kohavi, R., Provost, F.: Applications of Data Mining to Electronic Commerce. In Data Mining and Knowledge Discovery, Vol. 5 (2001) 5-10
5. Lent, B., Swami. A., Widom, J.: Clustering Association Rules. In Proc. 1997 Int. Conf. Data Engineering (ICDE'97), Birmingham, England (1997) 220-231
6. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York (1990)
7. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, California (1993)

Computational Lessons from a Cognitive Study of Invention

Marin Simina[1], Michael E. Gorman[2], and Janet L. Kolodner[3]

[1] EECS Department, Tulane University, New Orleans, LA 70118-5674
simina@eecs.tulane.edu

[2] TCC, SEAS, University of Virginia, Charlottesville, VA 22901
meg3c@virginia.edu

[3] College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280
jlk@cc.gatech.edu

Abstract. This paper investigates both the role of fine-grained historical cases in developing computational models of techno-scientific thinking and the impact of such models for supporting information search and further inventions and discoveries. In particular, we investigate Alexander Graham Bell's invention of the telephone and we propose a computational model to explain its essential aspects. We further derive lessons about how such model can be used to build human-computer interaction systems that augment the intelligence of users involved in information search. We conclude that historical data can be used to advance cognitive and computational theories of techno-scientific thinking and to build better human-information systems.

1 From the Telephone Invention to WWW Information Search

This paper investigates the lessons learned from developing computational models of techno-scientific reasoning based on fine-grained historical cases of invention. We highlight the applicability of such lessons for building intelligent systems to assist users in information search, since information search plays a significant role for pursuing new inventions and discoveries. We have chosen Alexander Graham Bell's invention of the telephone as our case study because it is one of the best documented and analyzed historical case of invention (e.g., Bruce, 1972; Bell, 1908).

Our investigation of Bell's inventions (Gorman, 1998, Simina et al., 1998) identified series of general criteria about scientific discovery and invention, which can be summarized as follows:

1. Invention and discovery depend on establishing that a problem is significant enough to be labeled an important achievement.
2. Invention and discovery depend on transforming that problem into a form that suggests a promising path to solution. This includes locating and transforming the necessary mechanical representations.
3. Invention and discovery depend on a combination of flexibility and stubbornness, depending on the cognitive styles and career trajectories of the inventors involved and on how they represent the problem.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 452-457, 2001.

4. Communication is part of the invention and discovery process.
5. Successful inventors and scientists often pursue networks of enterprises (open problems) that may interact among them. Such interactions are a major source of creativity.

The above general criteria also provide a framework for understanding other cases of invention and discovery. While Gorman (1998) shows how these criteria apply in the case of several scientists (such as Kepler, Lavoisier and Krebs), Simina (1999) proposes a computational model (ALEC) that simulates essential aspects of the telephone invention, which takes into account all the above criteria. The initial computational model (Simina et al., 1998) was based on an in depth analysis of Bell's inventions using case-based reasoning as an investigation tool. This analysis helped us to identify limitations of existing models of techno-scientific thinking (e.g. the inability to reason about several goals in parallel by taking advantage of opportunistic interactions among them). The initial version of ALEC addressed these limitations. Next, we used Gorman's criteria as additional constraints for refining our model. Simulating the invention of the telephone with ALEC helped us to refine Gorman's generalizations and their interpretation at a computational level. In turn, the resulting model can be used to better understand techno-scientific reasoning.

In the end, our computational model characterizes the long-term work of a creative reasoner in terms of partially-independent *enterprise goals*, high-level goals pursued in parallel that interact synergistically and evolve incrementally. While we agree that a reasoner explicitly addresses only one (current) goal at a time (e.g., Simon, 1989), we claim that the other (background) goals pursued in parallel, along with their (partial) solutions, provide a *reasoning context* for advancing solutions for a current goal. Moreover, if a reasoner is part of a team, then his reasoning context may include goals of other team members, and creative reasoning becomes distributed across the whole team.

Since our model highlights the role of information while pursuing enterprise goals, an interesting issue is whether it can be reused to support information search for future enterprises (e.g., invention or discovery). In what follows, we briefly present ALEC and then we describe Smart Agenda, a computational model for augmenting the intelligence of users involved in information search. Our objective is to investigate cognitive and computational models for supporting opportunistic information search on the WWW (and/or other large databases) by taking advantage of the lessons learned from Bell's case study.

2 ALEC

ALEC's functional architecture is presented in Fig. 1. According to our previous analysis (Simina et al. 1998, Simina 1999), a reasoner may internally pose its own enterprise goals (i.e., *Enterprise Posing*), or it may be interested in adopting externally-posed enterprise goals. After a new enterprise is posed, an *Enterprise Adoption* process must identify which of these posed enterprises are worth pursuing in the current context as active enterprises, which have to be postponed (by suspending them in

memory), and which should be ignored. A reasoner addresses an enterprise by concurrently evolving its specification and a pool of alternative solutions relying on his previous experience. This involves three processes: (1) *Evolve Specification*, (2) *Evaluate and Critique*, and (3) *Evolve Solutions*, which are together responsible for *Enterprise Processing.*

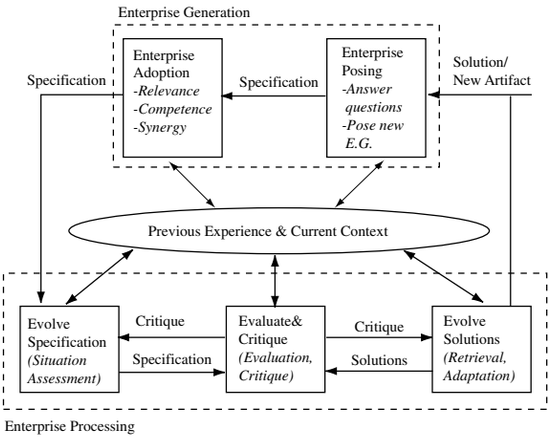


Fig. 1. ALEC: Functional Architecture

Each of the above processes makes inferences based on knowledge retrieved from the *Current Context* (knowledge and goals accessed recently) and/or from *Previous Experience*. The *Retrieval** algorithm is responsible for performing a fine-grain content-based retrieval from the Current Context, which simulates priming effects. If Current Context retrieval is unsuccessful, *Retrieval** performs an index-based retrieval from the Previous Experience repository (Kolodner, 1993). The index-based retrieval simulates free recall from long-term memory. To simulate opportunistic reasoning, suspended enterprise-goals can be indexed in the Previous Experience repository in terms of the missing knowledge that prevents finding solutions for the suspended goals. A detailed computational model can be found in Simina (1999).

The Enterprise Processing processes are directly relevant for information search. The next section presents Smart Agenda, a tool for supporting information search, which takes advantage of ALEC's opportunistic reasoning architecture.

3 Smart Agenda

Information search became a significant issue with the widespread reliance on the World Wide Web as an information delivery medium. Since existing deployed technologies for information classification and search (e.g., portals such as Yahoo) fail to provide the leverage needed to transform the massive amounts of new information into

knowledge (Furuta & Papakonstantinou, 1999), information search remains a significant method to access information in the WWW. Unfortunately, traditional search engines do not take into account many implicit aspects of the search process, such as the search context (e.g., the user's profile, the current problem that he is investigating and his other suspended goals), and consequently in many cases search engines fail to provide the right information at the right time. In such cases, people may take advantage of their previous experience to reformulate the initial query or to suspend the current information goal and resume it later, when relevant search knowledge becomes available. Intelligent search agents should be able to manifest similar adaptive behavior. If previous search experience is essential to evolve the information goals of a user (e.g., queries using a standard search engine), then the issue is how to capture this previous search experience and how to reuse it. Since the WWW is also a repository for a growing number of portals that manually encode search experience in limited domain, then a research hypothesis is that this manually encoded knowledge can be automatically captured by web agents, using methods similar with those described by Heflin & Hendler (2001), and reused to guide the evolution of the user's information goals. Such an approach takes advantage of the information that is already available on the web and transforms it into search knowledge. But there is no guarantee that the prerequisite knowledge for evolving an information goal is always available. In such situations people rely on opportunistic reasoning to suspend the current information goal and resume it later when information about how to evolve it becomes available (e.g., Simina 1999). The issue is how to discover that some current piece of information is relevant for a suspended goal. Then, a new search pattern connecting the suspended goal and the discovered information can be learned for future reuse. The second hypothesis is that opportunistic reasoning (see Figure 1) can help a reasoner to discover new search knowledge when traditional search methods fail. Beside providing a cognitive framework for understanding and supporting complex information retrieval, opportunistic information search provides a foundation for integrating existing tools developed for information retrieval.

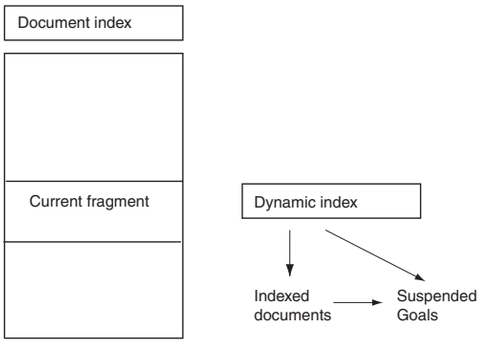


Fig. 2. Opportunistic information search

4 Related Work

Existing deployed tools for information retrieval include search engines, manually structured indices and knowledge bases (e.g., portals like Yahoo) and just-in-time information retrieval engines (JITIR; Rhodes & Maes, 2000). Each method has its own inherent limitations. Search engines may retrieve too many documents, most of them irrelevant to the current problem. The issue is how to reformulate the query to retrieve documents relevant to the current problem. Manually structured knowledge bases are designed to address well common queries, by suggesting some categories at every step of incremental query reformulation (e.g., Yahoo, Ask Jeeves). The issue is how to select a category for an unusual query. JITIR is a proactive technique that automatically retrieves documents relevant to the current task of the user. The user does not have to explicitly articulate his goal, and the JITIR engine may retrieve many documents that may be relevant for the current task or environment, but not necessarily for the implicit goal of the user.

One way to address the above limitations is to identify how people perform information search. Previous experience plays an important role in successful information search, and some researchers proposed cased-based systems that capture the expertise associated with information search (e.g., Jaczynski & Troussse, 1998; Leake et al., 2000). Such systems capture and store past navigation patterns of a group of users, and reuse past patterns that (partially) match the current search context to suggest what documents to examine next. However, in many cases it is not necessary to build navigation pattern libraries from scratch. Many existing web portals implicitly contain navigation knowledge and the only issues are: (1) to identify such portals and (2) to automatically extract navigation patterns and store them in a case library.

But what kind of support can be provided to a user when previous libraries of navigation patterns do not exist? People rely on opportunistic reasoning in such situations, i.e., they suspend the current information goal and they resume it when knowledge relevant to the suspended information goal becomes available. Previous research in opportunistic reasoning did not investigate information search. Just-in-time IR (Rhodes & Maes 2000) addresses only the issue of retrieving documents relevant for the current (implicit) information goal. A computational tool can keep track of the (suspended) information goals of a user and check opportunistically if the current document, relevant for the current information goal, is also relevant for any suspended goals. Currently we are experimenting with a prototype of Smart agenda built on top of the software described in Rhodes & Maes (2000).

5 Conclusions

Cognitive frameworks and methods applied to fine-grained historical case studies can add rigor to those analyses. Gruber (1974) proposed the framework of network of enterprises to explain Darwin's creativity. Gorman added his general criteria to Gruber's and showed how Gruber's analysis applies across invention and discovery cases (Gorman, 1998). But only with the addition of a computational model have we been

able to begin to understand the processing underlying goal suspension and the conditions for their reactivation (Simina 1999). This insight can now be applied both to understand fine-grained details of other historical case studies and to build intelligence augmentation tools (e.g., by supporting information search) to afford future inventions.

References

1. Furuta, R. and Papakonstantinou, Y. (1999). Information retrieval and data mining in the world-wide web, internet and wireless era. *1999 NSF Information and Data Management Workshop: Research Agenda for the 21st Century.*
2. Gorman, M.E. (1998). *Transforming nature: ethics, invention and discovery*. Kluwer Academic Publishers, Netherlands.
3. Gruber, H. (1974). *Darwin on Man: A Psychological Study of Scientific Creativity*. University of Chicago Press.
4. Heflin, J. and Hendler, J. (2001). A Portrait of the Semantic Web in Action. *IEEE Intelligent Systems*, March/April 2001.
5. Jaczynski, M. and Trousse, B. (1998). WWW assisted browsing by reusing past navigations of a group of users. In Cunningham, P.; Smyth, B.; and Keane, M., eds., Proceedings of the Fourth European Workshop on Case-basedReasoning, 160-171. Berlin: Springer Verlag.
6. Kolodner, J. (1993). *Case-Based Reasoning*. Morgan Kaufmann.
7. Leake, D.B., Bauer, T., Maguitman, A. and Wilson, D.C. (2000). Capture, Storage and Reuse of Lessons about Information Resources: Supporting Task-based Information Search. In *AAAI-2000 Workshop on Intelligent Lessons Learned Systems.*
8. Rhodes, B.J. and Maes, P. (2000). Just-in-time Information Retrieval Agents. In *IBM Systems Journal* 39 (3&4)
9. Schun, C.D. and Dunbar, K. (1996). Priming, analogy, and awareness in complex reasoning. *Memory & Cognition* 1996, 24(3), 271-284
10. Simina, M. (1999). *Enterprise-directed reasoning: Opportunism and deliberation in creative reasoning*. Ph.D. Thesis, Georgia Institute of Technology, College of Computing.
11. Simina, M., Kolodner, J., Ram, A. and Gorman, M. (1998). Opportunistic Enterprises in Invention. In *Proceedings of the Twentieth Conference of the Cognitive Science Society*, LEA.
12. Simon, H. 1989. The Scientist as Problem Solver. In Klahr, D. and Kotovsky, K., eds., *Complex Information Processing*, Lawrence Erlbaum Associates.
13. US v. Bell, A.G. 1908.*The Deposition of Alexander Graham Bell*. American Bell Telephone Company, Boston.

Component-Based Framework for Virtual Information Materialization

Yuzuru Tanaka and Tsuyoshi Sugibuchi

Meme Media Laboratory, Hokkaido University 060-8628 Sapporo, Japan
{tanaka, buchi}@meme.hokudai.ac.jp

Abstract. Various research fields in science and technology are now accumulating large amounts of data in databases, using recently developed computer controlled efficient data-acquisition tools for measurement, analysis, and observation. Researchers believe that such a huge extensive data accumulation in databases will allow them to simulate various physical, chemical, and/or biological phenomena on computers without carrying out any time-consuming and/or expensive real experiments. Information visualization for DB-based simulation requires each visualized record to work as an interactive object. Current information visualization systems visualize records without materializing them as interactive objects. Researchers in these fields develop their individual or community mental models on their target phenomena, and often like to visualize information based on their own mental models. We will propose in this paper a generic framework for developing virtual materialization of database records based on the component-ware architecture IntelligentBox that we developed in 1995 for 3D applications. This framework provides visual interactive components for (1) accessing databases, (2) specifying and modifying database queries, (3) defining an interactive 3D object as a template to materialize each record in a virtual pace, and (4) defining a virtual space and its coordinate system for the information materialization. These components are represented as boxes, i.e., components in the IntelligentBox architecture.

1 Introduction

Recently, extensive application of information technologies in various social activities such as production, distribution, sales, finance, communication, transportation, education, and welfare has enabled us to file large amounts of personal records in these social activities and to store them in databases. Although their use should not violate individual's privacy, they contain various useful knowledge resources that may not violate any privacy. Information visualization technologies as well as data mining technologies aim to support people to extract such knowledge resources. Most of the current information visualization systems propose various specific visualization schemes, assuming typical application fields and typical analysis methods in these fields. Some information visualization systems partially allow us to interactively define visualization schemes. These include Tioga 2 (Tioga DataSplash)[1], Visage[2], and DEVise[3]. They only allow us to make a selection out of *a priori* provided libraries of visualization schemes.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 458–463, 2001.

Various research fields in science and technology are now accumulating large amounts of data in databases, using recently developed computer controlled efficient data-acquisition tools for measurement, analysis, and observation. Researchers believe that such a huge extensive data accumulation in databases will allow them to simulate various physical, chemical, and/or biological phenomena on computers without carrying out any time-consuming and/or expensive real experiments. Here in this paper, we call such a new way of research in science 'data-based science'. Information visualization will be by no doubt one of the most powerful tools in data-based science. Current information visualization technologies, however, do not satisfy the requirements in data-based science.

Information visualization for DB-based simulation requires each visualized record to work as an interactive object. It should be easy enough for these researchers, who are not necessarily computer experts, to define the functionality of each visualized record as well as the spatial record arrangement. Current information visualization systems visualize records without materializing them as interactive objects. Instead of information visualization systems, we need an information materialization framework that allows us to materialize each record as an interactive visual object in a virtual space. Furthermore, researchers in these fields develop their individual or community mental models on their target phenomena, and often like to visualize information based on their own mental models. We need to provide these researchers with a new visualization environment in which they can easily define their own visualization schemes as well as various query conditions.

We will propose in this paper a generic framework for developing virtual materialization of database records based on the component-ware architecture IntelligentBox that we developed in 1995 for 3D applications. This framework provides visual interactive components for (1) accessing databases, (2) specifying and modifying database queries, (3) defining an interactive 3D object as a template to materialize each record in a virtual pace, and (4) defining a virtual space and its coordinate system for the information materialization. These components are represented as boxes, i.e., components in the IntelligentBox architecture.

2 IntelligentBox Architecture

IntelligentBox is a component-ware system for developing 3D interactive applications[4]. It is a 3D extension of a 2D meme media architecture IntelligentPad[5][6]. It calls components boxes. Boxes may have arbitrary internal functions as well as arbitrary 3D visual display functions. IntelligentBox provides a dynamic functional composition mechanism that enables us to geometrically and functionally combine 3D objects through direct manipulation on the screen to compose a complex 3D object. Only primitive component boxes need to be programmed by their developers. Composite boxes are also simply referred to as boxes unless this causes any confusion. Each box is logically modeled as a list of slots, each of which can be accessed by either a 'set' message or a 'gimme' message. Corresponding to each slot, a box has two procedures that are

respectively invoked by a 'set' and a 'gimme' message. In addition to these slots, each box has its properties such as its dimension, its orientation, and its angle. A box may define some of these properties as its slots, which allows other boxes to change those properties through their slot connection linkages to this box. A RotationBox, for example, has a cylinder shape, and rotates itself corresponding to user operations. It has a slot named #ratio whose value changes from 0.0 to 1.0 in proportion to its rotation angle. It rotates when the #ratio slot is set with a new value. Its direct manipulation changes not only its rotation angle but also its #ratio slot value.

A box can be connected to a single slot of no more than one other box. The former becomes a child box of the latter, while the latter is called a parent box of the former. Each child box is managed by the coordinate system defined by its master box. IntelligentBox allows us to make any box invisible. The child can access the connected slot of its parent by either a 'set' message or a 'gimme' message. A 'set' message takes one parameter, while a 'gimme' message has no parameter. The parent box can send an 'update' message to its child boxes. This message takes no parameter. In their default definitions, a 'set' message writes its parameter value into the corresponding slot register in the parent box, while a 'gimme' message reads the value of this slot register. An 'update' message tells the recipient that a state change has occurred in the parent box. In addition to these three standard messages, each box can accept geometrical messages such as 'resize', 'move', 'copy', 'hide', and 'show'.

3 Information Materialization through Query Composition

Our framework provides visual interactive components as boxes for (1) accessing databases, (2) specifying and modifying database queries, (3) defining an interactive 3D object as a template to materialize each record in a virtual pace, and (4) defining a virtual space and its coordinate system for the information materialization.

Figure 1 shows an example composition for information materialization. It specifies the above mentioned four functions as a flow diagram from left to right. The leftmost box is a TableBox, which allows us to specify a database relation to access; it outputs an SQL query with the specified relation in its 'from' clause, leaving its 'select' and 'where' clauses unspecified. The database is stored in a local or remote database server running an Oracle DBMS. When cliked, a TableBox pops up the list of all the relations stored in the database, and allows us to select one of them.

The second box is a TemlateManagerBox, which allows us to specify a composite box used as a template to materialize each record. It allows us to register more than one templates, and to select one from those registered for record materialization. When we select a template named t, the TemplateanagerBox adds a virtual attribute, 't' as TEMLATENAME, in the 'select' clause of the input query, and outputs the modified SQL query. The database has an addi-

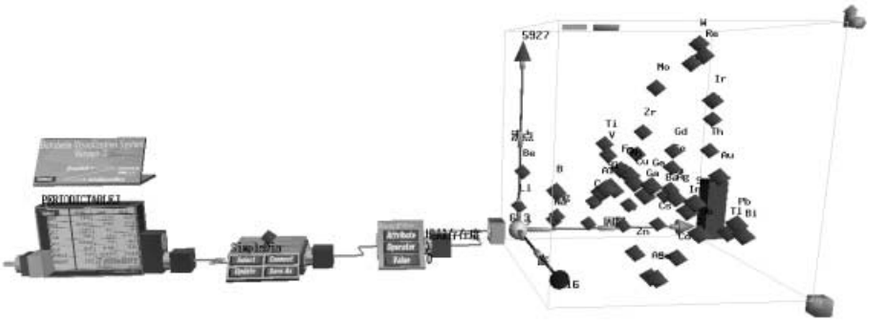


Fig. 1. An example composition for information materialization

tional relation to store the registered templates. This relation TEMPLATEREL has two attributes; TEMPLATENAME and TEMPLATEBOX. The second attribute stores the template composite box specified by the first attribute. In the later process, the specified SQL query is joined with the relation TEMPLATEREL to obtain the template composite box from its name. When we register a new template composite box, the TemplatManagerBox accesses the database DDD to obtain all the attributes of the relation specified by the input SQL query. It adds slots with these attributes to the base box of the template composite box. In the later process, the record materialization assigns each record value to a copy of this template box, which decomposes this record value to its attribute values and store them in the corresponding attribute slots of the base box.

The third component in the example is a RecordFilterBox, which allows us to specify attribute attr, a comparison operator θ, and a value v. This specification modifies the input query by adding a new condition attr θ v in its 'where' clause. The RecordFilterBox accesses the database DDD to know all the accessible attributes.

The last component in this example is a ContainerBox with four more components, an OriginBox, and three AxisBoxes. A ContainerBox accesses the database with its input query, and materializes each record with the template composite box. While an OriginBox specifies the origin of the coordinate system of the materialization space, each AxisBox specifies one of the three coordinate axes, and allows us to associate this with one of the accessible attributes. It also normalizes the values of the selected attribute. These two components also uses query modification methods to perform their functions.

In addition to the components used in the above example, the framework provides two more components, a JoinBox and an OverlayBox. A JoinBox accepts two input SQL queries, and defines their relational join as its output query. It allows us to specify the join condition. An OverlayBox accepts more than one

query, and enables a ContainerBox to overlay the materialization of these input queries. From the query modification point of view, it outputs the union of input queries with template specifications.

By using a ContainerBox together with an OriginBox and AxisBoxes as a template composite box, we can define a nested structure of information materialization as shown in Figure 2. The displacement of the origin of each record materializing ContainerBox from the map plane indicates the annual production quantity of cabbage in the specified year at the corresponding prefecture, while each record materializing ContainerBox shows the cabbage production changes during the last 20 years.

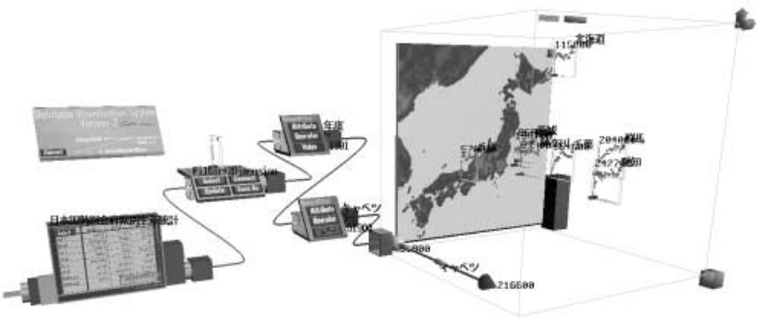


Fig. 2. A nested structure of information materialization.

As an application of our information materialization framework, we have been collaborating with Gene Science Institute of Japan to develop an interactive animation interface to access cDNA database for the cleavage of a sea squirt egg from a single cell to 64 cells. The cDNA database stores, for each cell and for each gene, the expression intensity of this gene in this cell. Our system that was first developed using our old information materialization framework without query modification components animates the cell division process from a single cell to 64 cells (Figure 3). It has two buttons to forward or to backward the division process. When you click an arbitrary cell, the system graphically shows the expression intensity of each of *a priori* specified set of genes. You may also arbitrarily pick up three different genes to observe their expression intensities in each cell. The expression intensities of these three genes are associated with the intensities of three colors RGB to highlight each cell of the cleavage animation. The wire-frame cube that encloses the whole egg performs this function. Keeping this highlighting function active, you can forward or backward the cell-division animation. The development of this system took only several hours using the geometrical models of cells that are designed by other people. The cDNA database is stored in an Oracle DBMS, which IntelligentBox accesses using Java JDBC.

We have applied our new information materialization framework to the same application. This extension enabled us to dynamically construct the same functionality within 15 minutes without writing any program codes or any SQL queries.

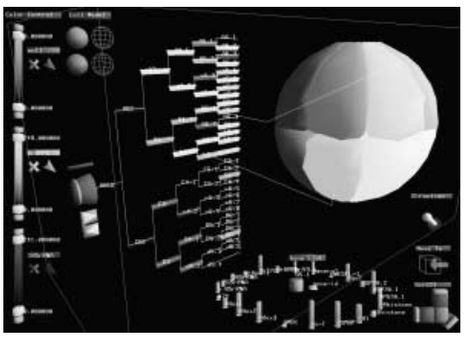


Fig. 3. Information materialization of the gene expression in the cleavage.

References

1. Alexander Aiken, Jolly Chen, Michael Stonebraker, and Allison Woodruff. Tioga-2: A Direct Manipulation Database Visualization Environment. Proceedings of the 12th International Conference on Data Engineering, pages 208-17, New Orleans, LA, February, 1996.
2. Derthick, M., Kolojejchick, J. A., and Steven Roth. An Interactive Visual Query Environment for Exploring Data. Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '97), pp 189-198, ACM Press, October 1997.
3. Miron Livny, Raghu Ramakrishnan, Kevin Beyer, Guangshun Chen, Donko Donjerkovic, Shilpa Lawande, Jussi Myllymaki, and Kent Wenger. DEVise: Integrated Querying and Visual Exploration of Large Datasets. Proceedings of ACM SIGMOD, May, 1997.
4. Y. Okada and Y. Tanaka. IntelligentBox:a constructive visual software development system for interactive 3D graphic applications. Proc. of the Computer Animation 1995 Conference, pp.114-125, 1995.
5. Y. Tanaka, and T. Imataki. IntelligentPad: A Hypermedia System allowing Functional Composition of Active Media Objects through Direct Manipulations. In Proc. of IFIP'89, pp.541-546, 1989.
6. Y. Tanaka. Meme media and a world-wide meme pool. In Proc. ACM Multimedia 96, , pp.175-186, 1996.

Dynamic Aggregation to Support Pattern Discovery: A Case Study with Web Logs

Lida Tang and Ben Shneiderman

Department of Computer Science
University of Maryland
College Park, MD 20720
{ltang, ben}@cs.umd.edu

Abstract. Rapid growth of digital data collections is overwhelming the capabilities of humans to comprehend them without aid. The extraction of useful data from large raw data sets is something that humans do poorly. Aggregation is a technique that extracts important aspect from groups of data thus reducing the amount that the user has to deal with at one time, thereby enabling them to discover patterns, outliers, gaps, and clusters. Previous mechanisms for interactive exploration with aggregated data were either too complex to use or too limited in scope. This paper proposes a new technique for dynamic aggregation that can combine with dynamic queries to support most of the tasks involved in data manipulation.

1. Introduction

Current technologies have enabled massive collections of data. Newer and faster algorithms for data analysis are always in demand to harness the flood. If the amount of data can be reduced to a manageable size, then humans can find patterns that automated algorithms may have missed. Dynamic Queries (DQ) is an interactive technique for data exploration [1]. Users manipulate sliders to filter out data. Each slider corresponds to an attribute of the data. A requirement of dynamic queries is that the visualization must keep up with the user's manipulation within 100 milliseconds. Since a large portion of the computer's computation is spent on visualization, when the datasets grow, the time to complete drawing grows proportionately. Thus DQ isn't suitable for dealing with large amounts of data.

Large datasets poses two problems to interactive exploration. One is how to represent the elements on the screen fast enough. Second is if you draw it on the screen, can the user even understand it. Visual occlusion is a problem in general for visualization. If the user can't see the data point, then the time spent drawing the item was wasted. This problem can be solved for small numbers of items. The commercial data analysis package, SpotFire (www.spotfire.com), randomly jitters the data points continuously, so that clusters that occupy the same point can be seen. With larger data sets, the occlusion problem grows even more pressing, due to the non-uniform nature of most data sets. The visual representation can deceive users by not showing clusters that exist in the data.

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 464-469, 2001.

Aggregation is an effective way of managing large data sets. It summarizes groups of similar data elements and can greatly reduce the number of glyphs that are shown on the screen. Because users can specify how to aggregate the data, the important aspects of the data will be preserved while the dataset size is reduced. Patterns that are hidden within millions of data points can emerge dramatically when aggregation reduces these into thousands of points. Fredrikson et al. [5] explored using aggregated data in conjunction with SpotFire, and demonstrated the uses of different kinds of aggregation with highway incident data. Hochheiser and Shneiderman [6] used aggregation to interactively explore web log data. In their study, the aggregation was done manually through SQL queries, though integration with an aggregation tool was suggested as a future direction.

2. Related Work

Putting aggregation and Dynamic Queries together in one interface is not a new idea. Goldstein et al. [2] proposed it in 1994. An interface mechanism called Aggregate Manager (AM) was combined with DQ, which produced a powerful combination (Figure 1). DQ is used to select a subset of the data set; this is transferred over to AM as an aggregate group. AM can then do aggregation on different aggregate groups, and pass the data back to DQ for display. This loop fulfills one of the lacking area of DQ: providing conjunct of disjunctive groups. Using AM along with DQ provides many possible combinations for data manipulation, which is powerful but can be hard for users to understand and fully control.

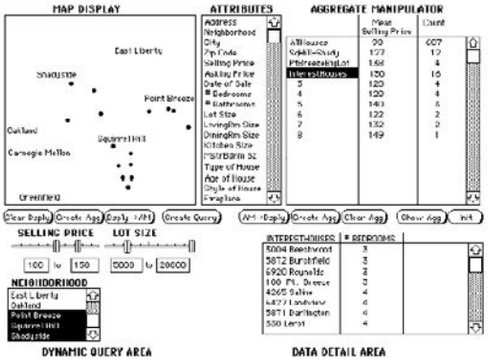


Fig. 1. The workspaces of AM with DQ

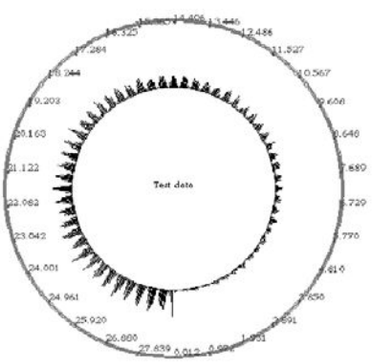


Fig. 2. SolarPlot showing a histogram

An alternative approach to user-controlled aggregation is automatic aggregation. Chuah and Roth [3] used automatic aggregation in SolarPlot, a circular histogram (Figure 2). Elements are mapped to a pixel on the circumference of a circle; the height of a spike that emanates from the pixel represents the number of data values that fall with in that pixel. This aggregation is intuitive and simple, the scale of the aggregation depends on the diameter of the circle, and the aggregated value is easily

understood. SolarPlot only encode one dimension of data in the visualization, thus any correlations between fields are harder to find.

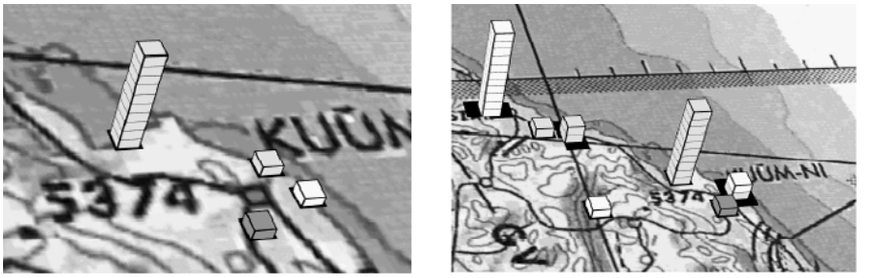


Fig. 3. Close up and zoomed out view of Aggregate Towers

Rayson's [4] Aggregate Towers provide another automatic aggregation interface. Data points are displayed as cubes on a 3d plane. As the user zoom in and out, data points are clustered based on their geospatial location (Figure 3). Stacks pointing out of the plane represent the aggregate groups. The cubes still retain their original color-coding. This automatic technique alleviates 2D occlusion problem by forcing it in to 3D. These stacks of data towers will occlude each other in 3D, but is easily remedied by allowing the user to freely rotate the view.

3. A Simple User Driven Aggregation Interface

Automatic aggregation is useful as a way to reduce occlusion. However, having no user control makes automatic aggregation of limited use for general datasets. Goldstein's AM is complex and hard to use because the user has complete control and no automation. Our system represents a middle of the road approach.

SpotFire's user interface was used as the starting point of our system. In SpotFire, a scatter plot of two attributes of the data is at center of the screen. Combo boxes at the edges of the axes select the fields being plotted. A panel on the right side displays DQ controls and detail on demand. The entire interface is in front of the user. Our system has similar characteristics as SpotFire. The aggregation controls are located on the left side so that DQ can be placed on the right side. The primary aggregation control is a combo box that can be enabled or disabled (Figure 4). Specifying a group of data manually is easy using DQ. However, creating many such groups can be time consuming and should be automated. The user only needs to select a field to group on, by using the "Group by" widget, and have the program sort out the groups. The default grouping algorithm used is equivalence grouping. For numerical data, equivalence is when they represent the same value, thus 4 and 4.0 are the same and belong in the same group. For categorical/string data, a case sensitive string comparison is used to determine equivalence, thus "4" and "4.0" as string are not the same. Should the user require a different grouping criterion, clicking on the "..." button to the right of the combo box will bring up an options dialog. Here, the user can choose which algorithm to use and to configure the algorithm to their liking. If the groups that are created are not specific enough for the user, they can be broken

down into subgroups. E.g. in the case of census data, we can group the entries based on gender, then subgroup based on age brackets, creating meaningful groups that can be used in aggregation. Subgroups are also controlled by checkable combo boxes. A combo box labeled "Subgroup by" will appear under the "Group by" widget after the user has selected a field to group by.

Once the grouping computation is finished, the results are shown on the screen with each dot now representing a particular group, the size of the dot is currently coded to show the number of elements in that group. The secondary aggregation controls are the aggregate method combo boxes. Those are located below the vertical axes field selector, and to the left of the horizontal axes field selector. The user can select different aggregation algorithms for each axis independently.

4. System Demonstration

The dataset used was extracted from web logs. The data is taken from University of Maryland's Computer Science web server. Only the requests that belonged to the HCIL section of the website (www.cs.umd.edu/hcil) were extracted. This is similar to the dataset that Hochheiser and Shneiderman [6] explored in their study. The data have the following five fields:

- Client host
- time: timestamp of request
- url: the URL requested
- return code: the server response code to request
- bandwidth used: number of bytes transmitted for that request

Web log data is very large and has only a few data fields. Traditional web analysis packages create tables of statistics and static graphs. The user merely feed the data to the program, and it is the program that decides what to report back to the user. Hochheister and Shneiderman argued in their paper that interactive star field visualization, like SpotFire, is a valid way of analyzing web log data. However, in order to find some of the interesting features involved preprocessing and aggregation. Thus, using the same web data will be a good test of the flexibility and power of our simple aggregation interface.

Since the web data consists of individual client requests, one logical grouping would be to group by user. By viewing the size of the groups, one can detect abnormally large numbers of requests from a particular user. We find that the Google spider the most frequent visitor of HCIL. To find out how much bandwidth Google consumed, we change the field we are viewing to "Bandwidth used" and set the aggregator function to sum the field (Figure 4). We found that it isn't Google, but another crawler, EoExchange that is using the most bandwidth. To view the access patterns of the clients, we can subgroup based on the time of access. Figure 5 shows access patterns of users over days. The bandwidth hog EoExchange shows up in this graph as well, while Google's accesses are well hidden and spread out across days.

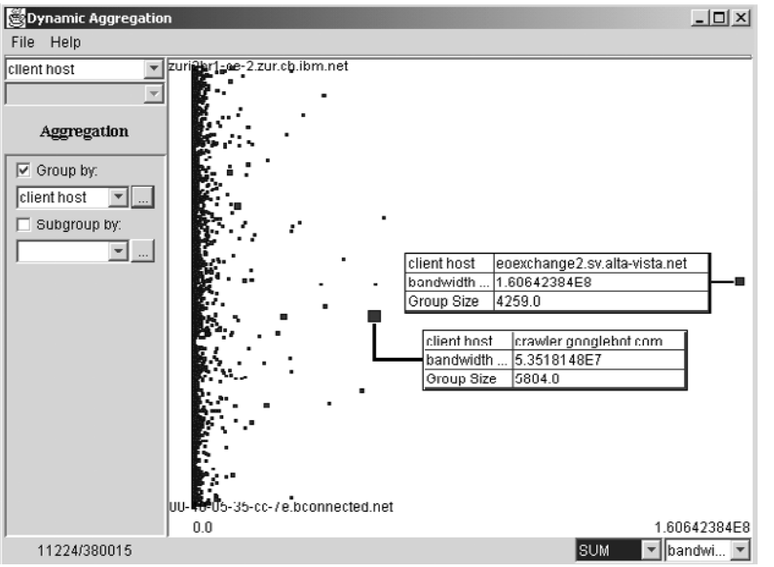


Fig. 4. Finding the most frequent visitor by aggregating by client host

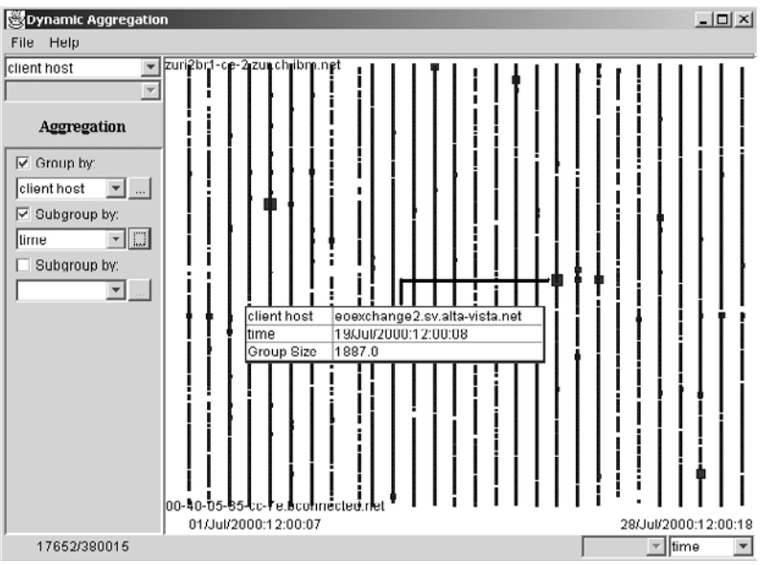


Fig. 5. User activity over days

5. Conclusion

We have developed simple manual aggregation interface that we believe the users can understand and use effectively. However, due to the inherent complicity of the aggregation concept, users should have in mind a specific question they would like answered. Unlike DQ, in which users can explore and experiment with data, aggregation should be thought of as creation of a new dataset. This new dataset can then be explored by DQ. A usability test should be conducted to test how readily users understand using the interface and which grouping algorithm and aggregation algorithm are needed to have a rich set of tools so the user can find answers to more complex questions than what was considered in the paper.

Acknowledgements

We thank Catherine Plaisant for her help in finding related work, Harry Hochheiser for help in getting the web log data, as well as the students of cmsc838b for support and inspiration.

References

1. Shneiderman, Ben. (1994). “Dynamic Queries for Visual Information Seeking.” *IEEE Software. 11(6), 70-77.*
2. Goldstein, Jade and Roth, Steven F. (1994) "Using Aggregation and Dynamic Queries for Exploring Large Data Sets" *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems 1994 v.2 p.200*
3. Mei C. Chuah and Roth, Steven F. (1998) "Dynamic Aggregation with Circular Visual Designs" *Proceedings of Information Visualization, IEEE, North Carolina, October 1998.*
4. Rayson, James K. (1999) "Aggregate Towers: Scale Sensitive Visualization Decluttering of Geospatial Data" *Proceedings of the 1999 IEEE Symposium on Information Visualization*
5. Fredrikson, A., North, C., Plaisant, C. and Shneiderman, B. (1999) “Temporal, Geographical and Categorical Aggregations Viewed through Coordinated Displays: A Case Study with Highway Incident Data” *Proceedings of the Workshop on New Paradigms in Information Visualization and Manipulation, Kansas City, Missouri, November 6, 1999 (in conjunction with ACM CIKM’99), ACM New York, 26-34.*
6. Hochheiser, H., and Shneiderman, B. (2001) “Using Interactive Visualizations of WWW Log Data to Characterize Access Patterns and Inform Site Design” *Journal of the American Society for Information Systems*, 52(4), February, 2001.

Separation of Photoelectrons via Multivariate Maxwellian Mixture Model

Genta Ueno[1], Nagatomo Nakamura[2], and Tomoyuki Higuchi[1]

[1] The Institute of Statistical Mathematics, Tokyo 106-8569, Japan
gen@ism.ac.jp
[2] Sapporo Gakuin University, Hokkaido 069-8555, Japan

Abstract. Electron velocity distribution obtained by direct spacecraft observation in space is contaminated by photoelectrons. The photoelectrons are generated due to the solar ultraviolet ray, and are regarded as artificial noise from a viewpoint of scientific research. We propose a method for separating photoelectron component from ambient electron component. Our method uses multivariate normal mixture model, whose parameters are determined via the Expectation-Maximization (EM) algorithm. Initial parameters of the EM algorithm are computed through the classification of the velocity space by a spherical surface of some arbitrary radius.

1 Introduction

The process of knowledge discovery begins with data acquisition and ends with identification of a new pattern in data. Between the start and the goal, the process includes various steps what we roughly call "data analysis." In the scheme proposed by Fayyad et al. [2], the process consists of (1) data selection, (2) data preprocessing, (3) data transformation, (4) data mining (hypothesis generation), and (5) hypothesis interpretation / evaluation. Creation and development of the computational strategy for such steps enable us to reduce the time in achieving the knowledge discovery, and are indispensable in dealing with large database. We have demonstrated that the multivariate normal mixture model is an effective tool for characterizing an observation of three-dimensional space plasma velocity distributions [6,7]. The normal distribution is called as the Maxwellian distribution in the plasma physical field. That is, when multiple peaks exist in the observed velocity distribution, these peaks can be well represented by the multiple Maxwellian distributions that compose the mixture model [4]. We applied the mixture model to the ion velocity distribution and determined the parameters of the model through the Expectation-Maximization (EM) algorithm [1,3]. This procedure is regarded as a step of "data mining" in the scheme of Fayyad et al.[2].

In this paper, we present that the similar procedure can be applied to the "preprocessing step" of the analysis of electron velocity distributions with minor modification. Since a spacecraft in sunlight is irradiated by the solar ultraviolet ray, photoelectrons are produced from illuminated surface material. The spacecraft is then charged to positive potential relative to the ambient plasma,

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 470–475, 2001.

which attracts the photoelectrons emitted from the surface. When the electron measurement is carried out in such an environment, returned photoelectrons are detected together with the ambient natural electrons what is originally expected to be observed. Since the amount of those photoelectrons is comparable or even larger than that of ambient electrons, it is difficult to obtain the real information from the data. This difficulty have prevented the progress of the quantitative study of electron dynamics.

However, we found that the two-component Maxwellian mixture model can represent the photoelectron and the ambient electron by the two component mixture model. While the algorithm used is similar to that in the previous work [6], a new algorithm has been developed in the part of setting the initial parameters.

2 Data

We used electron velocity distribution obtained by the Low Energy Particle Energy-per-charge Analyzer (LEP-EA) on board the Geotail spacecraft. LEP-EA measured three-dimensional velocity distributions by classifying the velocity space into 32 for the magnitude of the velocity, 7 for elevation angles, and 16 for azimuthal sectors (Figure 1).

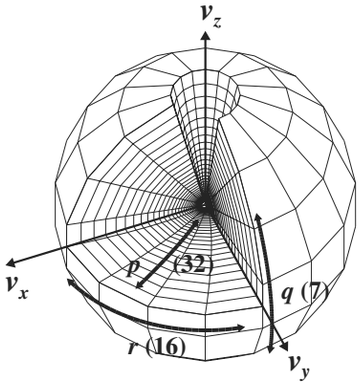


Fig. 1. Classes for observation of an electron velocity distribution with LEP-EA (RAM B mode)

We define a probability function of observed electron velocity $\boldsymbol{v}_{pqr}$ [m/s]:

$$f\left(\boldsymbol{v}_{pqr}\right) = \frac{f_0\left(\boldsymbol{v}_{pqr}\right) d\boldsymbol{v}_{pqr}}{\sum_{p,q,r} f_0\left(\boldsymbol{v}_{pqr}\right) d\boldsymbol{v}_{pqr}}, \tag{1}$$

where $f_0\left(\boldsymbol{v}_{pqr}\right)$ [$\mathrm{s}^3/\mathrm{m}^6$] is an observed electron velocity distribution function, and $d\boldsymbol{v}_{pqr}$ is the class interval whose class mark is $\boldsymbol{v}_{pqr}$. Subscription p, q and r

are indicators of the magnitude of the velocity, elevation angle, and azimuthal sector, and they take integers $p = 1, \cdots, 32$, $q = 1, \cdots, 7$, and $r = 1, \cdots, 16$.

3 Method

3.1 Multivariate Maxwellian Mixture Model and EM Algorithm

We approximate the probability function (1) by the mixture model composed of the sum of two multivariate Maxwellian distributions:

$$f\left(\boldsymbol{v}_{pqr}\right) \simeq \sum_{i=\mathrm{ph,am}} n_i g_i\left(\boldsymbol{v}_{pqr} | \boldsymbol{V}_i,\, \mathbf{T}_i\right), \tag{2}$$

where n_i is the mixing proportion of the Maxwellians ($\sum_{i=\mathrm{ph,am}} n_i = 1,\ \ 0 \leq n_i \leq 1$). Notations "ph" and "am" mean photoelectron and ambient electron, respectively. Each Maxwellian g_i is written as

$$\begin{aligned} & g_i\left(\boldsymbol{v}_{pqr} |\, \boldsymbol{V}_i,\, \mathbf{T}_i\right) \\ &= \left(\frac{m_e}{2\pi}\right)^{3/2} \frac{1}{\sqrt{|\mathbf{T}_i|}} \exp\left[-\frac{m_e}{2}\left(\boldsymbol{v}_{pqr} - \boldsymbol{V}_i\right)^T {\mathbf{T}_i}^{-1}\left(\boldsymbol{v}_{pqr} - \boldsymbol{V}_i\right)\right], \end{aligned} \tag{3}$$

where m_e [kg] is the electron mass, $\boldsymbol{V}_i$ [m/s] is the bulk velocity vector and $\mathbf{T}_i$ [J] is the temperature matrix of i-th Maxwellian. The log-likelihood of this mixture model becomes

$$l\left(\theta\right) = N \sum_{p,q,r} f\left(\boldsymbol{v}_{pqr}\right) \log \sum_{i=\mathrm{ph,am}} n_i g_i\left(\boldsymbol{v}_{pqr} | \boldsymbol{V}_i,\, \mathbf{T}_i\right), \tag{4}$$

where $\theta = (n_{\mathrm{ph}},\, \boldsymbol{V}_{\mathrm{ph}},\, \boldsymbol{V}_{\mathrm{am}},\, \mathbf{T}_{\mathrm{ph}},\, \mathbf{T}_{\mathrm{am}})$ denotes the all unknown parameters, and N is the total number of the particle count.

Partially differentiate (4) with respect to $\boldsymbol{V}_i$ and $\mathbf{T}_i^{-1}$ $(i = \mathrm{ph}, \mathrm{am})$ and put them equal to zero, we obtain the equations that should be satisfied by the parameters as maximum likelihood estimators. Utilizing these equations, we estimate the unknown parameters through the iteration of the EM algorithm with regarding posterior probabilities as unmeasured data [5,6]. We finish the iteration when the log-likelihood and unknown parameters become unchanged in the iteration.

3.2 Initial Parameters of EM Algorithm

To reduce an iteration of the EM algorithm, a proper setting for the initial parameters is desirable. We used the k-means algorithm for setting the initial parameters for the iteration of the EM algorithm in the previous work [6]. However, since the k-means algorithm is a clustering algorithm for an exclusive division, it is not applicable in setting initial parameters for a mixture model whose bulk velocities are close to each other.

Now we approximate an observation of electron distribution by photoelectrons and ambient electrons. The parameters are expected to satisfy

$$n_{\mathrm{ph}} > n_{\mathrm{am}}, \tag{5}$$

$$\boldsymbol{V}_{\mathrm{ph}} \simeq 0, \tag{6}$$

$$|\boldsymbol{V}_{\mathrm{am}}| < \sqrt{2\mathrm{tr}\,\mathbf{T}_{\mathrm{ph}}/3m_e} < \sqrt{2\mathrm{tr}\,\mathbf{T}_{\mathrm{am}}/3m_e}. \tag{7}$$

This is the case when the k-means algorithm does not work well. We then adopt the following method suitable for such a distribution.

1. Divide the 32 classes about the magnitude of velocity (radial direction in the velocity space) into two groups by a certain boundary of radius R ($R = 2, 3, \cdots, 31$).
2. Compute mixing proportion, bulk velocity vectors, and temperature matrices for both groups by usual moment calculation procedure.
3. Set these value as the initial parameters of the EM algorithm.

4 Application

The top panel of Figure 2 shows an observation of electron velocity distribution which was obtained in the time interval 1420:00–1420:12 on January 16, 1994. Displayed two lines are densities in the velocity space along the v_x and v_y axes. We find high density around the origin ($v_x = v_y = 0$), which corresponds to the photoelectrons. When applying the two-component Maxwellian mixture model to the data, we obtained photoelectron component and ambient electron component separately as shown in the bottom-left and bottom-central panels of Figure 2, respectively. The estimated parameter are given in Table 1. The sum of the two components are also shown in the bottom-right panel.

Table 1. Estimated parameters for the two-Maxwellian mixture model in the time interval 1420:00–1420:12 on January 16, 1994. The value of n is the mixing proportion multiplied by the total number density

Electron	n [/cc]	V_x [km/s]	V_y	V_z	T_{xx} [eV]	T_{xy}	T_{xz}	T_{yy}	T_{yz}	T_{zz}
photo	3.819	−534	257	123	7	0	0	7	0	7
ambient	0.064	270	−100	−87	144	−4	1	131	1	122

In setting the initial parameters of the EM algorithm, it may matter how to select the cutting radius R which potentially classify the data into photoelectron and ambient electron components. We found, however, that the results after the iteration of the EM algorithm are the same in most R selection. In this example, we can obtain the same result for $R = 2$ to 25.

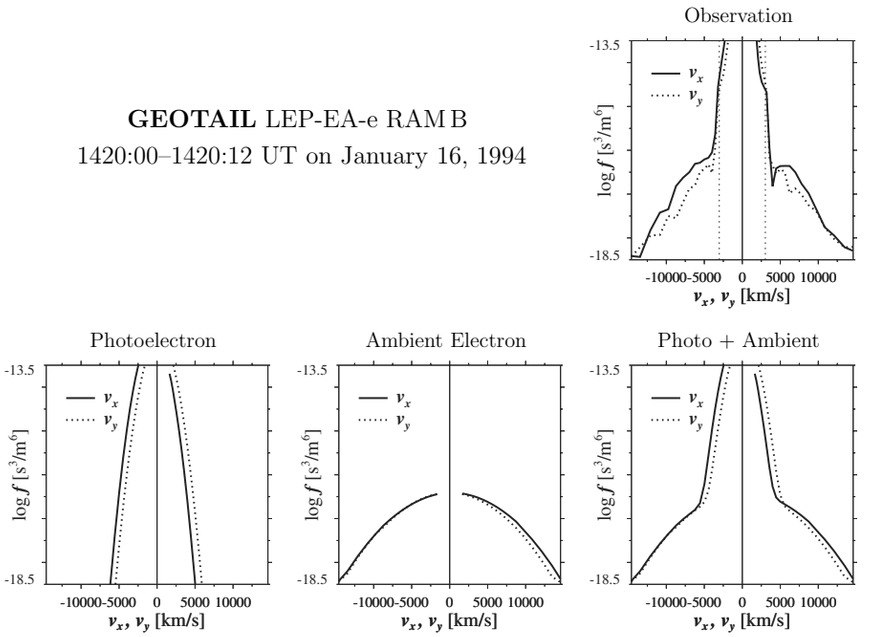


Fig. 2. Observation of electron velocity distribution along the v_x and v_y axes between 1440:00–1440:12 UT on January 16, 1994 (top). Bottom panels are estimated components for photoelectron (left), ambient electron (center), and the sum of the both (right). Two vertical broken lines in the top panel indicate electron velocities equivalent to the spacecraft potential at this time interval

5 Discussion

Traditionally, the spacecraft potential was utilized to decompose the electron data into photoelectron component and ambient electron component. When the spacecraft potential is ϕ [V], the equivalent electron speed $v_e = \sqrt{2e\phi/m_e}$ [m/s], where e [C] is the elementary electric charge. This means that a photoelectron particle whose speed is less than v_e cannot escape from the spacecraft and is pulled back to the spacecraft. Therefore, density of particles slower than v_e would be contributed by photoelectrons as well as ambient electrons. On the assumption that the photoelectrons were distributed within $|\boldsymbol{v}| \leq v_e$, they thus did not use the density of the slow particles and interpolated the density of slow speed particles by the density of particles faster than v_e. However, the equivalent electron speed v_e is not so accurate as an indicator of the photoelectron distribution. In the same time interval (1420:00–1420:12 UT on January 16, 1994), the spacecraft potential $\phi = 26.09\,\mathrm{V}$ and then $v_e = 3029\,\mathrm{km/s}$, which is shown in the top panel of Figure 2 as $\pm v_e$ by two vertical broken lines. The two lines are

located at smaller velocities than our expectation ($v_e \sim 4000\,\mathrm{km/s}$), and will give an inappropriate interpolation.

Since our method works automatically with less computational burden, it can compute the macroscopic quantity of the ambient electrons (n_{am}, $\boldsymbol{V}_{\mathrm{am}}$, and $\mathbf{T}_{\mathrm{am}}$) on board the spacecraft. It will be useful under the limited transmission resources from the spacecraft due to the telemetry constraint.

Acknowledgments. We would like to thank Prof. T. Mukai for providing us with Geotail/LEP data. This work was carried out under the auspices of JSPS Research Fellowships for Young Scientists.

References

1. Dempster, A. P., Laird, N. M., Rubin, D. B.: Maximum Likelihood from Incomplete Data via the *EM* Algorithm. J. Roy. Statist. Soc. B **39** (1977) 1–38
2. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery in Databases. AI Magazine **17** (1996) 37–54
3. McLachlan, G. J., Krishnan, T.: The EM Algorithm and Extensions. Wiley Series in Probability and Statistics. John Wiley and Sons, New York (1997)
4. McLachlan, G. J., Peel, D.: Finite Mixture Models. Wiley Series in Probability and Statistics. John Wiley and Sons, New York (2000)
5. Nakamura, N., Konishi, S., Ohsumi, N.: Classification of Remotely Sensed Images via Finite Mixture Distribution Models (in Japanese with English abstract). Proceedings of the Institute of Statistical Mathematics **41** (1993) 149–167
6. Ueno, G., Nakamura, N., Higuchi, T., Tsuchiya, T., Machida, S., Araki, T.: Application of Multivariate Maxwellian Mixture Model to Plasma Velocity Distribution Function. In: Arikawa, S., Morishita, S. (eds.): Discovery Science. Lecture Notes in Computer Science, Vol. 1967. Springer-Verlag, Berlin Heidelberg New York (2000) 197–211
7. Ueno, G., Nakamura, N., Higuchi, T., Tsuchiya, T., Machida, S., Araki, T., Saito, Y., Mukai, T.: Application of Multivariate Maxwellian Mixture Model to Plasma Velocity Distribution Function. to appear in J. Geophys. Res. (2001)

Logic of Drug Discovery: A Descriptive Model of a Practice in Neuropharmacology

Alexander P.M. van den Bosch

Groningen University, Center for Behavioral and Cognitive Neurosciences (BCN),
Department of Philosophy, A-Weg 30, 9718 CW, Groningen, The Netherlands
alexander@philos.rug.nl

Abstract. This paper reports on a logical model of the rational use of theory in a particular discovery problem in neuropharmacology, based on a case study of a practice of drug research for Parkinson's disease. This analysis describes how qualitative assumptions about the relation between properties of the nervous system are used to search for drug leads, *i.e.* properties for possible drug interventions. The logical structure of this drug lead discovery problem is briefly described together with the structure of some assumptions from the case study. It is briefly discussed how computational tools where used to explore these assumptions, and how they could possibly aid discovery in this domain.

1 Introduction

The study of scientific discovery is a subject that has a long tradition in philosophy of science and logic. The questions and methods in philosophy of science and logic usually focus on fundamental and normative matters that are often abstract and seem to lie far away from scientific practice. In one of the efforts of our department to bridge this gap I extensively studied a practice of neuropharmacology, in particular the drug research for Parkinson's disease conducted at the Groningen University Center for Pharmacy. Based on this study, where I interviewed and followed practitioners during their experiments, I modeled the logical structure of used theories and assumptions, and different types of problems that led to conceptual and empirical discoveries, *cf.* [1]. In this paper I first briefly describe the logical structure of rational drug lead discovery, one of the types of discovery problems I encountered. Secondly, I illustrate this problem with an example from the case study. I end with a discussion.

2 Rational Drug Lead Discovery

A main goal of drug research is to discover and design drugs and drug treatments. In the rational search for a drug treatment knowledge of biological processes is used to infer the effect of a drug intervention. A suggested intervention can either contain a description of a desired local influence of a drug on a biological system, or a description of a drug that is known to have the needed functional properties. These desired properties of a drug should cause a decrease in disease symptoms, and are

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 476-481, 2001.

called a *drug lead*, *cf.* [2]. The rational search for a drug lead can be described as a problem of qualitative reasoning. Knowledge of qualitative relations between variables describing properties of a pathological biological system can be sufficient to find variables that can influence that system in a desired way, *cf.* [3].

The search involved is structurally similar to that of explanatory or abductive reasoning, but has a different search goal. Instead of finding a simple hypothesis that explains an observed behavior, the task is to find a minimal intervention that has a desired effect on properties such as the behavior of the system, with minimal side effects. So, analogously to inference to the best explanation, this process can be called inference to the best intervention.

The object of drug treatment design does not initially concern the properties of a compound as in drug design, but the properties of a biological system, an organism. In the latter the goal is to create a drug so that it has given desired properties, in the former the goal is to create the behavior of a biological system so that it has given desired properties. These properties can be divided in structural and functional properties. A disease can be represented as a set of unwanted properties of a biological system. These can be compared with wished for properties of a system.

So we can define the characteristics of a disease as follows. Given the operational properties O(x) of a pathological system x and the wished for properties W, the characteristics of a disease y can be defined as $W \Delta O(x)$, the symmetric difference between O(x) and W:

$$W \Delta O(x) := W - O(x) \cup O(x) - W$$

The set O(x) contains all the considered properties of a system x, not only the pathological properties. So the set $W \cap O(x)$ is not empty. The goal of drug treatment is to change the properties O(x) of system x to O*(x) such that both $O^*(x) - W$ and $W - O^*(x)$ are minimized

Rational drug treatment discovery involves finding a drug treatment for a given pathological condition of a system by maximally employing known theories and knowledge about biological processes. A proper theory about a disease should be able to imply the pathological properties.

So, let a set H of theories about biological processes be given as well as background assumptions B(x) involved in the explanation of the observed properties among the properties O(x) of a pathological system x. The problem of the design of a drug treatment of the pathological properties $O(x) \Delta W$ is to cause only wished for properties from W by a drug intervention I(x) of the system, *i.e.* $H \cup B(x) \models I(x) \rightarrow W$. If a theory can imply the pathological condition, then we can use that knowledge to search for a suitable intervention, see Table 1.

Table 1. Logical structure of the rational drug lead discovery problem

Start :	$H \cup B(x) \models O(x)$
Goal :	$H \cup B(x) \models I? \rightarrow W$
Result :	I*(x)

The search goal is to find, by reasoning about processes represented in H, a proper drug intervention that influences processes that cause the desired properties W, but not those from O(x) – W. That is, the goal is to eliminate the difference between W and O(x). The result of the search is the suggestion of a manipulation of a local biochemical property that can be affected by a drug. A drug that has this wished for functional effect can be searched for in the set of known drugs, or pose a new problem for rational drug design.

Of course it would be ideal, given the known H and the nature of the disease, to infer a suggestion for a drug intervention I that only causes W. A drug usually also causes side effects, often creating undesired effects that are not part of the disease that is targeted. Therefore we need a gradual evaluation criterion for the improvement of suggestions, *cf.* [4]. Let us say that the moderated design goal is to find the suggestion I(x) such that its (predicted) consequence for a system, $H \cup B(x) \models I(x) \rightarrow P(x)$, resembles the desired condition W more than the pathological condition O(x), *i.e.* that: $P(x) \Delta W$ is a proper subset of $O(x) \Delta W$. That is, the drug should not have more unwanted consequences than accomplished desired consequences, *cf.* Fig. 1.

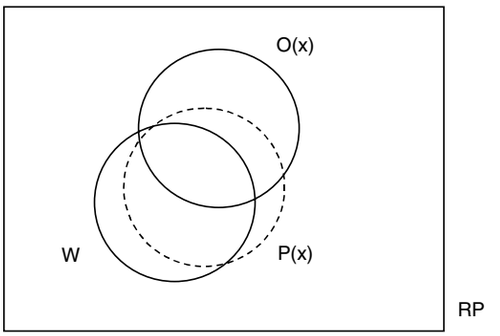


Fig. 1. Problem state in searching an intervention with effect P(x), in a space of relevant properties RP, that most resembles desired properties W in treating a pathological system x with operational properties O(x)

The evaluation of improvement of more than one drug suggestion can follow the same lines. A drug intervention I* of x is better than an intervention I if the properties of consequence P* resemble W more than those of P, *i.e.* $P^*(x) \Delta W$ is a proper subset of $P(x) \Delta W$.

However, this is only an evaluation of properties that is neutral to the different kinds of undesired properties. In this way an intervention could be inferred that treats most of the symptoms, but causes a symptom that is worse than the disease that is treated. This could be remedied by a ordering of the undesired properties, together with a measure of deviation.

The resulting suggestion for a drug intervention can on its turn be used to test the theories used to find the suggestion. Given an inferred drug intervention I(x), an experiment can be done and its resulting observation of the altered operational properties O(x) of x can be compared with the predicted properties P(x). A discrepancy can be used to redesign H, or the assumptions about B(x) or I(x).

3 Case Study: Drug Research for Parkinson's Disease

In Parkinson's disease patients suffer from motor behavior impairment. The cause of this disease is traced back to degeneration of a particular brain area called the *substantia nigra pars compacta* (SNC) that supplies the neurotransmitter dopamine to a brain area that is called the *basal ganglia*. Dopamine regulates the activation of the brain area called the *substantia nigra retaculata* (SNR), by excitating dopamine receptors of type D1 and inhibiting receptors of type D2. When the amount of dopamine depletes in Parkinson's disease, this balance is disrupted, resulting in a pathological increase of the activation of the SNR. The drug L-dopa, a precursor in the metabolism of dopamine, treats the disease. However, since this intervention acts on all dopamine receptors in the body it causes undesired side-effects. Currently selective drugs are searched for that only target a relevant subtype of the dopamine receptor in the basal ganglia. A problem for finding a treatment is to discover what subtypes to target. This search problem is logically reconstructed.

It is assumed that if a theory explains a proposition, then that theory should also be able to logically imply that proposition. A qualitative model of the basal ganglia was constructed that can logically imply the consequences of decreasing the amount of dopamine. Knowledge about the basal ganglia can be represented as a qualitative theory about a dynamical system, defined as a tuple $\langle V, Q, C \rangle$, where V is a set of variables which are reasonable functions over time, Q is a set of quantity spaces for those variables, and C is a set of constraints on variables in V. For the basal ganglia theory I used two basic variables describing firing rate (*f*) of nerve cells in a cell group, nuclei or pathway, and the amount (*a*) of a particular neurotransmitter released in the vicinity of a cell group, nuclei or neural pathway. The constraint relation $y = M^{+/-} x$ is used to state that the change of values of y over time is positively /negatively monotonically related to the change of value of x.

Figure 3.2 displays a fragment of the basal ganglia model H_{BG}, containing of the cell nuclei called striatum, SNC, Gpe, SNR, and the neurochemicals L-Dopa, dopamine, GABA, and glutamate, for further details see [1] and [5]. For example, the increase of the firing rate of the SNC causes an increase in the amount of dopamine in the striatum, while this latter increase causes a decrease in activation of the neural pathway that signals to the Gpe, propagating further through the neural circuitry. Given the model it can be deduced that a decrease of the amount of dopamine implies an increase of the firing frequency of the SNR, *i.e*:.

$$H_{BG} \cup B: \{a(\text{DA, striatum}) = \text{dec}\} \models P: \{f(\text{SNR}) = \text{inc}\}$$

The use of qualitative models of biological processes could help to explain and find suggestions for possible treatments. The discovery problem for a treatment can be defined as a search in a solution space of conceptually possible interventions. We start with a qualitative model and known initial values of its variables. A goal of desired variable values is set. Reasoning backward from the goal values one can explore possible manipulations of the variables. The approximation criterion, as defined in the former section, can then be used to measure the difference between the goal values and the values caused by a particular manipulation, implementing a means-end analysis.

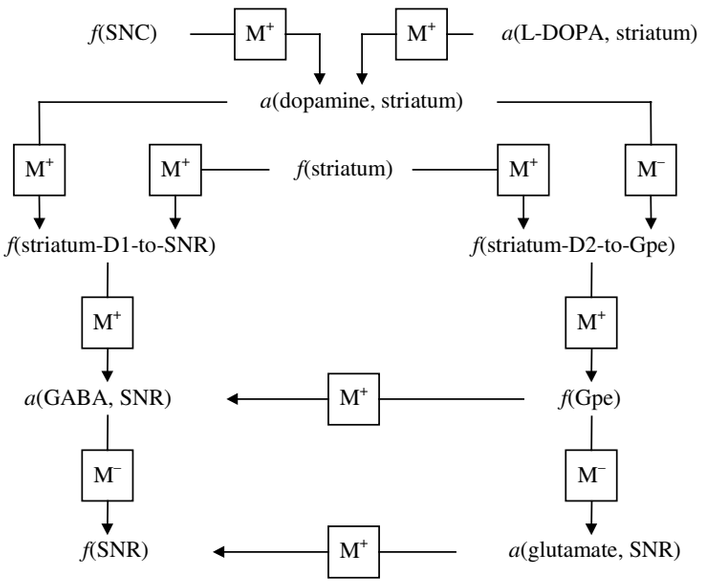


Fig. 2. A part of H_{BG}, a qualitative model of some assumptions about the *basal ganglia*

In Parkinson's disease, the goal set includes a lower activation frequency of the SNR than in the pathological case. A search through possible manipulations will not only find an increase of the amount of L-dopa in the striatum. It will also find that a decrease of the firing rate of the indirect pathway between the striatum and the GPe results in a decrease of the firing rate of the SNR. Administering a selective D2 agonist can cause such a decrease, with a lesser effect on dopaminergic pathways in other parts of the body than the effect of L-Dopa. These kinds of suggestions were on its turn used to empirically test the basal ganglia model, *cf.* [1].

4 Discussion

This logical reconstruction tells us nothing new about what to do about Parkinson's disease. Yet by making the knowledge and reasoning explicit (by describing it formally) it is possible to increase the complexity of models that are used in practice, such as that of the basal ganglia. Via a computer program as a modeling tool it is possible to keep track of, and further investigate, all the consequences of such a model. The conceptual space of the basal ganglia model was explored using the qualitative simulator QSIM *cf.* [6], and GARP, a general architecture for reasoning about physics, *cf.* [7], yet both do not implement abductive discovery operators. An implementation of abduction and inference to the best intervention in the context of this domain is part of our ongoing research.

However, the bigger problem to make such tools useful in practice is the availability of biological theory in a formal representation. It would be ideal if scientists in biology and medicine would publish their results both in natural language and in a formal format. It would already provide a much clearer view on results if it would be qualitatively stated whether investigated parameters where found to be positively or negatively related. Different publications about a domain taken together would provide a search space that could be relatively manageable, leaving the details of testing discovered interesting hypotheses to further empirical research based on such suggestions.

References

1. van den Bosch, A.P.M.: Rationality in Discovery - A Study of Logic, Cognition, Computation and Neuropharmacology, Institute for Language Logic and Computation, Amsterdam (2001)
2. Vos, R.: Drugs looking for diseases. Innovative drug research and the development of the beta blockers and the calcium antagonists. Kluwer Academic Press, Dordrecht (1991)
3. van den Bosch, A.P.M.: Qualitative Drug Lead Discovery. In working notes of the International Congress on Discovery and Creativity, Ghent, (1998) 163-165
4. Kuipers, T.A.F., Vos, R. & Sie, H.: Design Research Programs and the Logic of their Development. Erkenntnis (37), (1992) 37-63
5. van den Bosch, A.P.M.: Inference to the Best Manipulation - a case study of qualitative reasoning in neuropharmacy. In Foundations of Science 4 (4). Special issue on Scientific Discovery and Creativity: Case studies and computational approaches. Guest editors: J. Meheus & T. Nickles (1999) 483-495
6. Kuipers, B.: *Qualitative Reasoning, Modeling and simulation with incomplete knowledge*. Cambridge, MA: MIT Press (1994)
7. Bredeweg, B.: *Expertise in Qualitative Prediction of Behaviour.* Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands, (1992)

SCOOP: A Record Extractor without Knowledge on Input

Yasuhiro Yamada[1], Daisuke Ikeda[2], and Sachio Hirokawa[2]

[1] Graduate School of Information Science and Electrical Engineering,
Kyushu University, Fukuoka 812-8581, Japan
yshiro@matu.cc.kyushu-u.ac.jp
[2] Computing and Communications Center,
Kyushu University, Fukuoka 812-8581, Japan
{daisuke,hirokawa}@cc.kyushu-u.ac.jp

Abstract. We present a record extractor system SCOOP. We assume that semi-structured documents given to SCOOP contain similar formats and each of them has only a record consisting of some different fields. SCOOP treats a document as just a string and does not use knowledge on input except that a field is surrounded with delimiters, a left delimiter ends with ">", and the corresponding right delimiter begins with "<". By counting substrings, SCOOP roughly divides into two parts: contents of the fields and others. SCOOP counts substrings near boundaries of two parts and extracts the most frequent substrings as delimiters. We show experimental results with news articles written in English or Japanese. A record consists of the headline and the body text on this experiment. SCOOP extracts records at a high rate.

1 Introduction

The number of Web pages is extremely increasing. These pages contain useful data. Nevertheless the structure of the data is described in some pattern of strings and not explicates compared to data in database systems. That is why we call them semi-structured documents [1]. A major target of Web mining is a set of semi-structured documents.

An important application of Web mining is extraction contents of semi-structured document as records. A record is a basic notion of database systems. A database consists of records, and a record consists of some fields. A field is the minimum unit in a database. To use Web pages like a database system, one needs wrappers that extract contents of the pages. In this paper, we describe a system that extracts contents of Web pages as records without any knowledge on input documents.

An input for our system is a set of semi-structured documents which contain similar formats such as Web pages in the same site, or pages generated automatically with search facility. There are two types of the problem for record extraction depending on the number of records contained in an input file [2]. In the single instance problem, input is a file which contains many instances of

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 482–487, 2001.

the same record. In the multiple instance problem, input is a set of files each of which contains an instance of the same record. We consider the multiple instance problem.

A format of Web pages is usually described with some patterns of HTML tags. The record structure of Web pages is written in such a way that each field is packed with a pair of tag sequences as the left parenthesis and the right parenthesis. In [3], Atzeni and Mecca implemented a language in which one can describe a wrapper by specifying these tag patterns. In [6,7], Kushmerick, Weld and Doorenbos applied machine learning techniques for extraction of these tag sequences. In [8], Sakamoto, Arimura, and Arikawa proposed a wrapper attentioned to the tree structure of HTML documents. Most of these approaches require some instances of records for learning or use some knowledge such as the type of used tags. For example, the wrapper in [7] has to know the position of records. In [4], Embley, Jiang and Ng showed some boundary detection techniques. But they assumed that the boundaries are determined by the tags hr, td, tr, a, table, p, br, h4, h1, strong, b and i.

We do not use such instances or knowledge. We treat a document as just a string. All knowledge on input is that a left delimiter ends with ">" and the corresponding right delimiter begins with "<". Most of the field boundaries experimentally found are substrings of tag sequences such as "t>" and "<hr><".

We assume the following five heuristics to guess structures of HTML files: frequent strings are not the contents of fields [5], each field is surrounded with two substrings of tag sequences, instaces of the same field are surrounded with the same pair of delimiters, each file contains an instance of the same record, and a left delimiter ends with ">" and the corresponding right delimiter begins with "<".

SCOOP has three steps: (1) SCOOP uses `FindOptimal` developed in [5] to divide roughly into two parts by counting substrings: contents of the fields and others. `FindOptimal` assumes that frequent substrings express structures of documents and are not the contents of fields. (2) SCOOP counts substrings near boundaries of two parts and extracts the most frequent substrings as delimiters. (3) The delimiters provide for SCOOP to extract fields. This is based on the very simple idea to count frequent substrings twice, but SCOOP extracts records at a high rate as given news articles written in English or Japanese.

2 SCOOP System

SCOOP is a system which extracts records from semi-structured documents with similar formats and outputs a list of records (see Fig. 1). A pair of delimiters surrounding each field is called a *rule*. A left delimiter is called a *start_string* and the corresponding right delimiter is called an *end_string*.

On Preprocessing of Fig. 1, SCOOP utilizes the algorithm `FindOptimal` developed in [5]. `FindOptimal` also receives a set of semi-structured documents

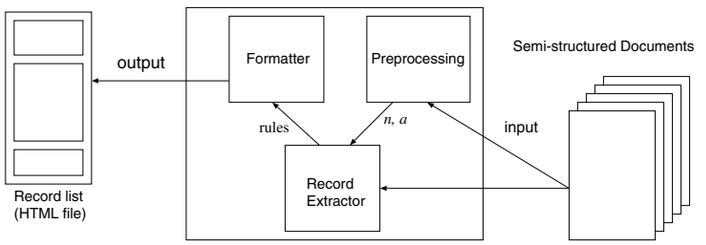


Fig. 1. The outline of SCOOP system

with similar formats and divides roughly into two parts: contents of the fields and others. `FindOptimal` also treats input documents as just strings.

`FindOptimal` outputs a pair (n, a), where n denotes a length and $0 \leq a \leq 100$ denotes a percentage. Consider that all substrings with length n of input documents are sorted by the number of their occurrences in the decreasing order. If a substring with length n is in the top a-percent of the sorted list, then we say that the substring is frequent on (n, a). We put gray color on frequent substrings of each input string like $ac\mathit{cba}\mathit{acb}c$, where $accbaacbc$ be a part of an input string, and ba and cb are frequent on some pair. `FindOptimal` assumes that frequent substrings express structures of documents and are not the contents of fields. So black substrings cover with contents of the fields and gray substrings cover with others.

`FindOptimal` finds (n, a) which attains a locally minimum alternation count. An alternation count is the number of alternations between black and gray substrings. In [5], it is experimentally shown that, given news articles written in English or Japanese, `FindOptimal` divides into the contents of the fields and others with more than 95% accuracy.

This is the very high accuracy, but it is not complete. In this paper, we adjust an output of `FindOptimal` according to the followings: a black substring with length less than n is treated as a gray substring, a gray sequence of tags with length less than n among black is treated as a black substring, and a black substring is treated as a gray substring if the black substring is in a tag and is surrounded with gray substrings. SCOOP treats black substrings as fields on adjusted output.

SCOOP receives an adjusted output and finds all rules surrounding each black substring on Record Extractor of Fig. 1. SCOOP counts substrings with length 2 surrounding each black substring. SCOOP increases the length of this substring by one if the most frequent substring is not unique on each document. It continues this until an extracted the most frequent substring becomes to be unique on each document.

First, SCOOP finds “>” (the end of a start_string) which appears just before the first black substring in each file. SCOOP counts 2-length substrings ending with this “>”. SCOOP increases the length of this substring by one if

the most frequent substring with length 2 is not unique on each document. And let the most frequent substring be start_string if it becomes to be unique on each document.

Next, SCOOP similarly finds end_string which appears just after black substring and starts from "<" (the start of a end_string). And let this start_string and the corresponding end_string be a candidate of the first rule. But if the number of files from which strings are not extracted by this rule is more than the half of the input files, SCOOP does not use this candidate. SCOOP proceeds the black substring which appears in the next of end_string until it does not extract a more rule.

Finally, on Formatter of Fig. 1, SCOOP extracts strings inside each rule and outputs HTML file as a list.

3 Experiments

We implement SCOOP in Perl, and execute non-parallelly on Compaq Alpha Server GS320 (731MHz Alpha21264) on several sets of inputs. An input for SCOOP is a set of news articles in the same language, with similar formats and not including noises. An article we use is written in English or Japanese. It is provided as an HTML file which has a headline and a body text.

First, we use articles obtained from "Los Angeles Times[1]" as English news articles. All pages linked from the URL are collected and then non-article pages are removed manually. The number of the articles in "Los Angeles Times" is 150 files. The total size is about 4.7M Bytes. SCOOP extracts three fields, the first field is the title of each page, the second field is the headline and the third field is the body text. The contents of the first field equal to those of the second one and are surrounded with title tags. SCOOP extracts all fields from 150 files (100%). Fig. 2 is a part of an HTML file outputed by SCOOP.

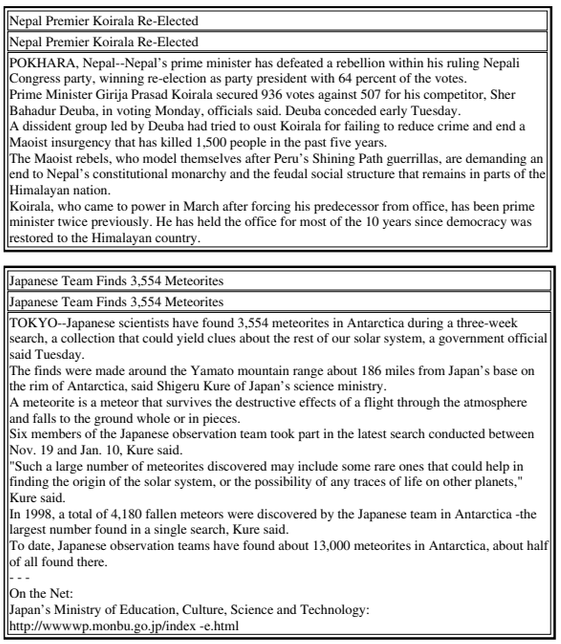
SCOOP preserves tag sequences in a field. For example, there are some "<br>" (means breaking a new line) in articles of "Los Angeles Times", and SCOOP preserves them as shown in Fig. 2. This means that SCOOP only finds tags designating static structures.

We expect that fields are only the headline and the body when we watch an HTML file used in the experiment. But SCOOP outputs that the title of each page is also a field. On the other experiments, there are some cases that SCOOP extracts some fields which we do not expect.

Next, we use articles obtained from "Yomiuri On-Line[2]" as Japanese news articles. The number of the article in "Yomiuri On-Line" is 65 files. The total size is about 1.4M Bytes. SCOOP extracts two fields, the first field is the headline and the second field is the body text. SCOOP extracts the first and second field from 165 files (100%).

On the other experiments, SCOOP fails to extract fields from several files. When SCOOP finds delimiters of a field, it begins at the boundary of black

[1] http://www.latimes.com/
[2] http://www.yomiuri.co.jp/



Nepal Premier Koirala Re-Elected

Nepal Premier Koirala Re-Elected

POKHARA, Nepal--Nepal's prime minister has defeated a rebellion within his ruling Nepali Congress party, winning re-election as party president with 64 percent of the votes.
Prime Minister Girija Prasad Koirala secured 936 votes against 507 for his competitor, Sher Bahadur Deuba, in voting Monday, officials said. Deuba conceded early Tuesday.
A dissident group led by Deuba had tried to oust Koirala for failing to reduce crime and end a Maoist insurgency that has killed 1,500 people in the past five years.
The Maoist rebels, who model themselves after Peru's Shining Path guerrillas, are demanding an end to Nepal's constitutional monarchy and the feudal social structure that remains in parts of the Himalayan nation.
Koirala, who came to power in March after forcing his predecessor from office, has been prime minister twice previously. He has held the office for most of the 10 years since democracy was restored to the Himalayan country.

Japanese Team Finds 3,554 Meteorites

Japanese Team Finds 3,554 Meteorites

TOKYO--Japanese scientists have found 3,554 meteorites in Antarctica during a three-week search, a collection that could yield clues about the rest of our solar system, a government official said Tuesday.
The finds were made around the Yamato mountain range about 186 miles from Japan's base on the rim of Antarctica, said Shigeru Kure of Japan's science ministry.
A meteorite is a meteor that survives the destructive effects of a flight through the atmosphere and falls to the ground whole or in pieces.
Six members of the Japanese observation team took part in the latest search conducted between Nov. 19 and Jan. 10, Kure said.
"Such a large number of meteorites discovered may include some rare ones that could help in finding the origin of the solar system, or the possibility of any traces of life on other planets," Kure said.
In 1998, a total of 4,180 fallen meteors were discovered by the Japanese team in Antarctica -the largest number found in a single search, Kure said.
To date, Japanese observation teams have found about 13,000 meteorites in Antarctica, about half of all found there.
- - -
On the Net:
Japan's Ministry of Education, Culture, Science and Technology:
http://wwwwp.monbu.go.jp/index -e.html

Fig. 2. A part of output of SCOOP given Los Angeles Times. This is a part of PS file generated by "Netscape"

substrings in an output of `FindOptimal`. If many contents end with "`</tagA> </tagB>`" and other contents end with "`</tagB>`", SCOOP can not extract contents end with "`</tagB>`" because SCOOP decides that end_string is "`</tagA> </tagB>`". Therefore, the accuracy for some contents extraction is lower. But, we think if we use some knowledge of HTML, SCOOP can extract such contents at a high rate on this case.

When `FindOptimal` is given files generated with search facility as input, `FindOptimal` extracts contents at a low rate. Some substrings in fields are frequent on these files, for example, query terms appear frequently in the result of search. So `FindOptimal` can not extract such substrings as the contents of fields. In such a case, SCOOP can not extract records. SCOOP is influenced by the accuracy of `FindOptimal`.

4 Conclusion

We implemented SCOOP system which extracts records from semi-structured documents with similar formats and outputs them as a list. We experimented with news articles written in English or Japanese. SCOOP extracted records at a

high rate although SCOOP does not use knowledge on input. Moreover SCOOP extracted a field which we had not expected.

SCOOP assumes that frequent substrings express structures of documents and are not the contents of fields. And SCOOP extracts the most frequent substrings around black substrings as delimiters.

If semi-structured documents given to SCOOP include noise, SCOOP extracts incorrect rules and can not extract records. Thus, it is an important future work to guarantee noise-tolerance for SCOOP. And if semi-structured documents given to SCOOP have some instances of the same field on each document, SCOOP extracts as instances of different fields and can not extract as instances of the same fields. Thus, it is also an important future work for SCOOP.

References

1. S. Abiteboul, P. Buneman and D. Suciu, Data on the Web. Morgan Kaufmann Publishers, 2000.
2. N. Ashish and C. Knoblock, Wrapper Generation for Semi-structured Internet Sources. Proc. Workshop on Management of Semistructured Data, 1997.
3. P. Atzeni, G. Mecca, Cut and Paste. Proc. the 16th ACM SIGMOD Symposium on Principles of Database Systems, 144–153, 1997.
4. D. W. Embley, Y. Jiang and Y. -K. Ng, Record-Boundary Discovery in Web Documents. Proc. ACM SIGMOD Conference, 467–478, 1999.
5. D. Ikeda, Y. Yamada and S. Hirokawa, Eliminating Useless Parts in Semi-structured Documents using Alternation Counts. Proc. the 4th International Conference on Discovery Science, Lecture Notes in Artificial Intelligence, 2001. (to appear)
6. N. Kushmerick, D. S. Weld and R. B. Doorenbos, Wrapper Induction for Information Extraction. International Joint Conference on Artificial Intelligence, 729–737, 1997.
7. N. Kushmerick, Wrapper Induction: Efficiency and Expressiveness. Artificial Intelligence Vol. 118, 15–68, 2000.
8. H. Sakamoto, H. Arimura and S. Arikawa, Extracting Partial Structures from HTML Documents, Proc. the 14th International FLAIRS Conference: Knowledge Discovery and Data Mining. (to appear)

Meta-analysis of Mutagenes Discovery

Premysl Zak[1], Pavel Spacil[2], and Jaroslava Halova[2]

Academy of Sciences of The Czech Republic
[1] Institute of Computer Science, Pod vodarenskou vezi 2, CZ 182 07 Prague 8, Czech Republic
zak@cs.cas.cz

[2] Institute of Inorganic Chemistry, CZ 250 68 Rez near Prague, Czech Republic
halova@iic.cas.cz

Abstract. The meta-analysis of the challenging data set on the mutagenicity of nitroaromatic compounds has been performed. There are two ways of structure coding: standard topological indexes or so-called fingerprint descriptors. In our previous work, a unique structure coding by fingerprint descriptors was used for the discovery of mutagenes with GUHA+/- software system. GUHA can process nominal variables, which are transformed to binary strings in the course of computation. Any structure coding can then be used for GUHA. The data encoded by topological indexes were processed by GUHA+/- software system as well. The hypotheses on the reasons for mutagenicity of nitroaromatic compounds were generated by GUHA+/- for Windows. Processing of data encoded by topological indexes was rather demanding because of the large number of structure descriptors. Meta-analysis by combining fingerprint descriptors for a posteriori structure templates resulting from previous analyses and more flexible topological indexes seems to be more appropriate.

1 Meta-analysis

The aim of meta-analysis is to relate the performance of different machine-learning algorithms on the characteristics of data set [1]. The famous mutagenicity data set [2] represents the discovery challenge tackled by many researches. Muggleton et al [3] used Inductive Logic Programming [ILP] system Progol for mutagene discovery with [2] data subset. This subset was already known not to be amenable to statistical regression, though its complement was adequately explained by the linear model [2]. In [3] topological descriptors were used. The advantage of Muggleton's approach is its flexibility. Inokuchi et al [4], [5] used the principle of graph abduction for this data set. Inokuchi's approach is similar to our fingerprint descriptors coding. Inokuchi's approach is more flexible mining frequent graph substructures similar to our a priori fingerprint descriptors. On the other hand, the computation based on fingerprint descriptors is faster in the order of magnitude. We propose connection of both methods for generating fingerprint descriptors by graph abduction. Okada obtained important results using the Cascade Model [6]. The results of Okada's approach are in accordance with our results obtained both by the fingerprint and topological

K.P. Jantke and A. Shinohara (Eds.): DS 2001, LNAI 2226, pp. 488-492, 2001.

descriptors mentioned below. Matsuda et al [7] apply Graph Based Induction Learning technique to extract typical patterns from graph data.

2 Principles of GUHA Method

Basic ideas of GUHA (General Unary Hypotheses Automaton) method were presented in [8] already in 1966. Starting notion of the method is an object. Each object has properties expressed by variables ascribed to this object. For example object can be a man with properties given by the variables of sex, age, color of eyes, etc. In order to make reasonable knowledge discovery we need to have a set of objects of the same kind, which differ in values of variables defined on them.
The aim of GUHA method is to generate hypotheses on relations among the properties of the objects, which are in some respect interesting. This generation is processed systematically; the machine generates in some sense all the possible hypotheses and collects the interesting ones. The hypothesis is generally composed of two parts: from the antecedent and the succedent. The antecedent and the succedent are tied together by the generalized quantifier, which describes the relation between them. The antecedents and succedents are propositions on the object in the sense of the classical propositional logic, so they are true or false for particular object. These propositions can be simple or compound similarly to propositional logic. Compound propositions (literals) are usually composed of conjunction connective. Formulation of these propositions is enabled through original variable categorization. Given an antecedent and succedent, the frequencies of four possible combinations can be computed and expressed in compressed form as the so-called four-fold table (ff-table). General ff-table looks like this:

ff-table	Succedent	Non(succedent)
Antecedent	a	b
Non(antecedent)	c	d

Where “a” is the number of the objects satisfying both the antecedent and succedent, “b” is the number of the objects satisfying the antecedent but not the succedent, etc.
A generalized quantifier is a decision procedure assigning 1 or 0 to each ff-table. If the value is 1, then we accept the hypothesis with this ff-table, if it is 0, then we do not accept it. The basic Fisher generalized quantifier defined and used in GUHA is given by Fisher exact test known from mathematical statistics. For each hypothesis, value of Fisher statistic given by values a, b, c, and d of ff-table is computed. Its value, simply said, describes the measure of association between the antecedent and succedent. The lower the value of Fisher quantifier is, the better association is.
In [9] information content of rules obtained by mining procedure is proposed, which suggests a promising improvement of the procedure.

3 Data Preprocessing

Mutagenicity data set was given in two tables. Both data sets can describe compounds in the same manner, therefore there can be redundancy in the data. This redundancy is unpleasant in the search for quantitative structure-activity relationships (QSARs), but the used method (GUHA) enables the choice of the best of redundant variables for dependency relation.
All descriptors seem to be cardinal and so their preprocessing is necessary. We divided each variable into several intervals. Among the variables, there is a huge amount of features and indexes, which are mostly unknown to us, so we divided them into three intervals equifrequently (Low, Medium, High) automatically. That means, one interval – one variable category - involves about 75 cases. We omitted the hypotheses with medium activity from the interval $(-0.1,1.9\rangle$ in the succedent.
Some variables include only one value (0), and they cannot be useful anyhow and therefore they were omitted (a_nP, Fcharge, ...). Furthermore, the data could be used directly as the input of GUHA+/-.
Meta-data were input as additional fingerprint descriptors.

4 The Results of Data Mining

GUHA is used for generating hypotheses of the following type:

"if the car is black and is cheaper than 50000 crowns, then the owner is a widower older than 50."

Most of variables were nominal or dividable into natural intervals. Now, the task is not only to find the hypotheses of the type:

"LUMO from x to y causes Activity from xx to yy."

Such results can be substantially dependent on the interval division of variables. Therefore, we should try to find the variable (combination of variables) affecting mutagenicity.
A correlation matrix of all variables was computed and hypotheses consisting of redundant variables were omitted (only the best of them were chosen).
Our efforts were divided into four phases. Fisher quantifier was always used as the lead criterion in the search for hypotheses. The second important criterion was Prob (number of cases fulfilling the hypothesis divided by the number of cases fulfilling the antecedent) that characterized hypotheses in terms of an implication.

Most of the hypotheses refer to Activity of the "High" value interval. For example the best hypotheses can be interpreted in the following manner:

1. *Presence of **XVIII structure fragment** [6] increases the probability of **Activity** in the "High" interval.*

2. **6-5-6** ***condensed rings*** *and **LUMO** "High" (high reduction potential) increase the probability of **Activity** in the "High" interval (in agreement with [6] and [7].*

3. *More than one **NO_2 groups** in minimum **tricyclic compound** (I_I = 1 [2]) increase the probability of **Activity** in the "High" interval.*

4. *Presence of **XVIII structure fragment** [6] and the absence of mutagenes from [3] increase the probability of **Activity** in the "High" interval.*

The most interesting hypothesis is the following, undoubtedly. This hypothesis has excellent both characteristics (Fisher and Prob). We could say, that it is the best hypothesis, we have generated, at all.

5. *"High" **balabanJ index** (based on molecular graph distance index) and **Polarity** "Low" increase the probability of **Activity** in the "Low" interval.*

5 Conclusion

Chemical interpretation of the most favorite hypothesis is the following:
High Balaban's connectivity topological index (based on molecular graph distance index) and low polarity implies low mutagenicity.

Apart from this hypothesis representing new toxicological knowledge, several hypotheses on the reasons of mutagenicity (mutagenes) were generated using GUHA method. Some of them represent new toxicological knowledge. Other hypotheses are in accordance with toxicological evidence. [10]

We presented a number of hypotheses discovered by GUHA+/-. The next step should be studying these hypotheses and generating more precise hypotheses including three or more variables in antecedent, in accordance with knowledge of the variables.

Our assumption that GUHA can be used in the search for interdependencies seems to be right. We tried to draw dependency graphs of the best hypotheses and they showed the trends.

According to the theory of global interpretation of multiple hypotheses testing the global significance of our results was considered. From this point of view the results as a whole can be interpreted as sufficiently reliable knowledge on the universe of which the data form a random sample.

References

1. Todorowski, L., Dzeroski, S.: Experiments in Meta-level Learning with ILP, In: J.M.Zytkow, J. Rauch (Eds): Proceedings of The Third European Conference on Principles of Data Mining and Knowledge Discovery, PKDD'99, Prague 1999 (Prague School of Economics), Lecture Notes in Computer Science, LNCS 1704, Springer Verlag Berlin, Heidelberg, New York, Tokyo 1999
2. Debnath, A.K., Lopez de Compadre, R.L., Debnath, G., Schusterman, A.J., Hansch, C.: Structure Activity Relationship of Mutagenic Aromatic and Heteroaromatic Nitro Compounds. Correlation with molecular orbital energies and hydrophobicity , Journal of Medicinal Chemistry, 34(2) (1991) 786
3. Muggleton, S., Srinivasan, A., King, R.D., Sternberg, M.J.E.: Biochemical Knowledge Discovery Using Inductive Logic Programming In: Motoda, H., Arikawa, S.,: (Eds.) : Proceedings of The First International Conference on Discovery Science, Lecture Notes in Computer Science , LNCS 1532, pp. 291-302 , Springer Verlag Berlin, Heidelberg, New York, Tokyo (1998)
4. Inokuchi, A. et al.: Applying Algebraic Mining Method of Graph Substructures to Mutagenesis Data Analysis, In. Suzuki E. (Ed): International Workshop of KDD Challenge on Real-world Data, 4th. Pacific Asia Conference on Knowledge Discovery and Data Mining, Kyoto, 2000
5. Inokuchi , A. Washio, T., Motoda, H.: An Apriori Algorithm for Mining Frequent Substructures form Graph Data, In Zighed, D.A., Komorowski, J., Zytkow, J. (Eds) Proceedings of The Fourth European Conference on Principles of Data Mining and Knowledge Discovery, PKDD 2000, Lyon, Lecture Notes in Computer Science, LNCS 1910, Springer Verlag Berlin, Heidelberg, New York, Tokyo 2000
6. Okada, T.: SAR Discovery on the Mutagenicity of Aromatic Nitro Compound Studied by the Cascade Model, In. Suzuki E. (Ed): International Workshop of KDD Challenge on Real-world Data, 4th. Pacific Asia Conference on Knowledge Discovery and Data Mining, Kyoto, 2000
7. Matsuda, T., Horiuchi, T., Motoda, H., Washio, T.: Graph-Based Induction for General Graph Structure Data and Its Application to Chemical Compound Data. In: Arikawa, S., Morishita, S., (Eds. Proceedings of the Third International Conference on Discovery Science, Kyoto 2000, LNCS 1967 Springer Verlag Berlin, Heidelberg, Tokyo.
8. Chytil, M., Hajek, P., Havel, I.: The GUHA method of automated hypotheses generation, Computing, 293-308, 1966
9. Smyth, P., Goodman, R. M.: An Information Theoretic Approach to Rule Induction From Databases. IEEE Transactions on Knowledge and Data Engineering 4(4)(1992) 301.
10. Balaban, A.I., Chiriac, A., Motoc I., Simon, Z.: Steric Fit in Quantitative-Structure Activity Relationships, Springer Verlag, Berlin 1980

Author Index