ICDM '02

The 2002 IEEE International Conference on Data Mining

International Workshop on Active Mining (AM-2002)

Sponsored by the IEEE Computer Society

Maebashi TERRSA, Maebashi City, Japan

December 9, 2002

International Workshop on Active Mining (AM-2002)

Preface

Recent advancement made through extensive studies and real world applications reveals that no matter how powerful computers are now or will be in the future, KDD researchers and practitioners must consider how to manage ever-growing data which is, ironically, due to the extensive use of computers and ease of data collection, ever-increasing forms of data which different applications require us to handle, and ever-changing requirements for new data and mining target as new evidences are collected and new findings are made. In short, the need is ever increasing in this era of information overload for 1) identifying and collecting the relevant data from a huge information search space, 2) mining useful knowledge from different forms of massive data efficiently and effectively, and 3) promptly reacting to situation changes and giving necessary feedback to both data collection and mining steps.

Active mining follows a spiral model of scientific discovery in spirit. It is a collection of activities each solving a part of the above need, but collectively contributing to the various mining need in a spiral way. This workshop is organized by the members of "Active Mining" project, which is a four year project funded by the Japanese Ministry of Education, Culture, Sports, Science and Technology starting in September 2001. The project aims at exhibiting the interleaved and spiral effect of the above three steps by challenging to analyze medical and chemical dataset as a common test bed.

The aim of this workshop on Active Mining, held in Maebashi on 9 December 2002, was to collect and report the experience and the new methodologies gained through different application areas with the above needs in mind, and offer an opportunity for researchers in different countries to meet and share the ideas.

This workshop consists of three parts: three invited talks, 6 contributed talks and 16 poster presentations. Luc de Raedt (Albert-Ludwigs-University Freiburg, Germany), Stefan Wrobel (Fraunhofer AIS & University of Bonn, Germany) and Saso Dzeroski (Jozef Stefan Institute, Slovenia) have generously accepted our invitation. In this proceedings, within each part, papers are ordered alphabetically by the last name of the first author.

Each contributed paper (both aural and poster) was reviewed by at least three program committee members and volunteer reviewers. The organizers wish to express their sincere thanks to those who reviewed the papers for their detailed and constructive comments and suggestions that were very useful for improving the quality of the papers, as well as to the above mentioned three invited speakers for their impressive and valuable talks.

Papers are also available from "http://www.ar.sanken.osaka-u.ac.jp/activemining/am2002.html"

Workshop Chair — Hiroshi Motoda
Program Chair — Takashi Washio

December 2002

Table of Contents

Poster Papers

On Molecular Feature Mining and Inductive Databases (Extended Abstract)

Luc DE RAEDT Christoph HELMA Stefan KRAMER Sau Dan LEE

Inst. für Informatik
University of Freiburg
Georges Koehler Allee 79
D-79110 Freiburg, Germany
deraedt@informatik.uni-freiburg.de

Abstract

This paper first introduces the molecular feature miner MOLFEA, *a domain specific inductive database.* MOLFEA *is then taken as the starting point for discussing various aspects of inductive databases.*

1. Introduction

Ever since the start of the field of data mining, it has been argued that data mining should be tightly integrated with databases. More recently, the framework of inductive databases has been proposed by Iemielinski and Mannila [5]. Inductive databases allow the user not only to query the data that resides in the database but also the patterns that hold among them. Thus in the inductive database framework, data mining becomes an interactive querying process.

Because we are still far away from a generally accepted theory of inductive databases, it is important to develop domain specific inductive databases and to gather experiences with these systems. In this spirit, we present the molecular feature miner MOLFEA [7], in which one can mine for fragments of interest in sets of molecules. This is realized using an inductive query language in which the user can declaratively specify the constraints that interesting fragments should satisfy. Given an inductive query, the inductive database management system computes all fragments that satisfy the constraint.

MOLFEA is then used as the starting point for discussing various design issues, technical developments as well as challenges for inductive database systems. This includes the use of border representations (version spaces), data structures, as well as query execution and optimization issues.

2. MOLFEA

MOLFEA is a domain specific inductive database for mining features of interest in sets of molecules. The examples in MOLFEA are thus molecules, and the patterns are molecular fragments. More specifically, in [7] we employed the 2D structure of molecules, and linear sequences of atoms and bonds as fragments.

An example molecule named AZT, a commonly used drug against HIV, is illustrated in Figure 1. Two interesting molecular fragments discovered using MOLFEA are:

```
'N=N=N-C-C-C-n:c:c:c=O'
'N=N=N-C-C-C-n:c:n:c=O'
```

In these fragments, `'C'`, `'N'`, `'Cl'`, etc. denote elements[1], and `'-'` denotes a single bond, `'='` a double bond, `'#'` a triple bond, and `':'` an aromatic bond. The two fragments occur in AZT because there exist labeled paths in AZT that corresponds to these fragments.

The type of fragment employed in MOLFEA is more limited than that used in the work by Inokuchi *et al.* [9], who use subgraphs as fragments. On the other hand, MOLFEA's query language is more powerful because Inokuchi *et al.* focus on minimal frequency thresholds only.

3. An inductive query language

We now introduce an inductive query language for mining molecular features. We allow for queries q that either contain no variables or contain exactly one variable τ as in $q(\tau)$. Queries without variables will be interpreted as true or false; queries $q(\tau)$ with one variable τ will be interpreted as sets $\{i \mid q(i)\ is\ true\}$.

- Let g and s be fragments. Then g *is more general than* s, notation $g \preceq s$, if and only if

[1] Elements involved in aromatic bonds are written in lower-case.

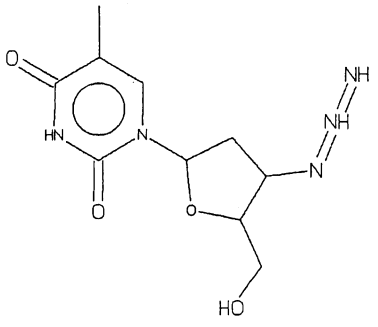


Figure 1. Chemical Structure of Azidothymidine

g is a subfragment (i.e. substring) of s. E.g. `'N=N=N-C'` and `'C-N=N=N'` are more general than `'N=N=N-C-C-C-n:c:c:c=O'`.

- The $\preceq$ relation can now be used in primitive constraints of the type $p \preceq p', \tau \preceq p, \neg(\tau \preceq p), p \preceq \tau$, and $\neg(p \preceq \tau)$, where τ denotes the target pattern and p and p' specific patterns. E.g. $\tau \preceq$ `'C-N=N=N'` yields as solutions the set of subfragments of `'C-N=N=N'`.

- Let p be a pattern and D a data set, i.e. a set of examples. Then $\mathit{freq}(p, D) = \mathit{card}\{e \in D \mid p \preceq e\}$, where $\mathit{card}(S)$ denotes the cardinality of the set S. So, $\mathit{freq}(p, D)$ denotes the number of instances in D covered by p, i.e. the frequency of p in D.

- The $\mathit{freq}(p, D)$ construct can now be used in constraints of the following form: $\mathit{freq}(p, D) \geq t$, $\mathit{freq}(p, D) \leq t$, $\mathit{freq}(\tau, D) \geq t$ and $\mathit{freq}(\tau, D) \leq t$ where t is a numerical threshold, τ is the queried pattern, and D is a data set.

- Given that we work with sets it will be useful to employ traditional set operations: $i \in I, i \notin I, \tau \in I$ as well as $\tau \notin I$ where τ is the queried pattern, i an element and I is a set.

- Finally, the language $\mathcal{IL}$ consists of any boolean expression b involving the above introduced primitives. We allow for the usual boolean connectives $\wedge, \vee, \neg$.

So, within the sketched language, we can formulate the following queries:

- All traditional set operations can be performed; e.g. the query $(\tau \in D_1) \vee (\tau \in D_2)$ denotes the set $D_1 \cup D_2$.

- Traditional minimal frequency queries can be performed using, e.g. $\mathit{freq}(\tau, D_1) \geq 2$.

- One can also obtain the set of examples in D_3 covered by a given pattern p_2 using the query $(\tau \in D_3) \wedge (p_2 \preceq \tau)$.

- Complex queries such as $(\mathit{freq}(\tau, D_{pos}) \geq n) \wedge (\mathit{freq}(\tau, D_{neg}) \leq m)$ ask for the set of patterns that are frequent on the positive examples D_{pos} and infrequent on the negatives in D_{neg}. This type of query has been successfully used in a database containing molecules tested against HIV, cf. [7]

Notice that even though the inductive query language is simple, the range of queries that can be expressed is quite large. The data mining primitives are an extension of those employed in the current implementation of the MOLFEA system for mining molecular features [7]. The key differences are that MOLFEA does not yet support the set oriented primitives and that it only allows for conjunctive queries. Here, arbitrary boolean queries are supported.

The present query answering mechanism in MOLFEA is based on the levelwise version space algorithm [2], which integrates Agrawal et al.'s [1] with Mellish's [8] description identification algorithm. More formally, let P be a pattern set, and define $max(P) = \{p \in P \mid \neg\exists q \in P : p \preceq q\}$, i.e. $max(P)$ contains the maximally specific elements in P, and define $min(P)$ dually, i.e. $min(P) = \{p \in P \mid \neg\exists q \in P : q \preceq p\}$, i.e. $min(P)$ contains the maximally general or minimally specific elements in P. We can then also define the borders $S(P)$ and $G(P)$ of a pattern set P as $S(P) = max(P)$ and $G(P) = min(P)$. The interesting point about borders is that they can be used as condensed representations of the solution set. It has been shown [2] that queries q that are a conjunction of anti-monotonic and monotonic constraints are completely characterized by the sets $S(q)$ and $G(q)$, i.e. $q = \{p \mid \exists s \in S(q), g \in G(q) : g \preceq p \preceq s\}$. These sets can be computed using the levelwise version space algorithm. A constraint c is *anti-monotonic* (resp. monotonic) w.r.t. generality whenever

$$\forall \mathit{patterns}\ s, g : (g \preceq s) \wedge (s \in \mathit{sol}(c)) \rightarrow (g \in \mathit{sol}(c))$$

(resp. $(g \in \mathit{sol}(c)) \rightarrow (s \in \mathit{sol}(c))$). Anti-monotonic (resp. monotonic) constraints have the property that whenever a pattern s satisfies the constraint, all its generalizations (resp. specializations) will also satisfy the constraint. The basic anti-monotonic constraints in our framework are: $(\tau \preceq p), \mathit{freq}(\tau, D) \geq m$, the basic monotonic ones are $(p \preceq \tau), \mathit{freq}(\tau, D) \leq m$. Furthermore the negation of a monotonic constraint is anti-monotonic and vice versa.

4. Extensions and Challenges

Whereas our initial results on MOLFEA stated that the solution set of a conjunctive query involving anti-monotonic and monotonic constraints can be represented

using a single version space, our more recent results [3] state that the answer set of any boolean query over anti-monotonic and monotonic constraints can be represented using as the union of different version spaces. To see why this is the case, rewrite the query in a disjunctive normal form. Each conjunction with the disjunctive normal form will then involve monotonic and anti-monotonic constraints. Hence, the earlier result applies and the conjunction can be represented as a version space, and the original set as the union of such version spaces. This result in turn leads to some interesting questions such as "What is the minimum number of version spaces needed to represent the answers to an inductive query?". This last question is partially answered in [4].

A second crucial issue in inductive databases concerns the data structures for representing pattern sets. To this aim, we have developed version space trees (for string patterns) [4], which combine principles of suffix trees with those of version spaces. The version space tree w.r.t. a conjunctive inductive query can efficiently be computed using variants of standard data mining algorithms. Furthermore, version space trees can easily be transformed into the border set representations mentioned above and vice versa. In addition, recognizing whether a pattern belongs to a pattern set represented by a version space tree can be decided in linear time.

A third important question concerns the optimization of inductive queries. More specifically, given an arbitrary boolean inductive query, what is the most effective way of computing the pattern set satisfying the query? To answer this question, it is important to exploit the correspondence between logical connectives (such as $\wedge$, $\vee$ and $\neg$) and (pattern) set operations (such as $\cap$, $\cup$ and complement). If operations on pattern sets are well-defined and efficiently computable, then the answers to a boolean inductive query (e.g., $p \vee (q \wedge r)$) can be computed using the corresponding operations on the pattern sets (e.g., $P \cup (Q \cap R)$). Operations on pattern sets in the form of version spaces and version space trees are currently being studied, cf. e.g. [3]. These operations are in a sense analogous to the operations in the relational algebra. Taking this analogy with relational query answering and optimizations further it may be possible to optimize inductive queries by studying various (logically equivalent) reformulations of a given inductive query and selecting that with minimal cost.

Finally, w.r.t. the application domain of molecular feature mining, we are studying various extensions of the linear molecular fragments. These are concerned with the use of tree and graph structured fragments (cf. also [9]), the introduction of 3D fragments and the use of the discovered fragments for the induction of structure activity relationships [6].

Acknowledgements

This work was partly supported by the European IST FET project cInQ. The authors are grateful to Heikki Mannila and Manfred Jaeger for their contributions to some of the theoretical aspects sketched in the last Section of this paper.

References

[1] R. Agrawal, T. Imielinski, A. Swami. Mining association rules between sets of items in large databases. In *Proc. SIGMOD*, pp. 207-216, 1993.

[2] L. De Raedt, S. Kramer. The level wise version space algorithm and its application to molecular fragment finding. In *Proc. IJCAI*, 2001.

[3] L. De Raedt. Query evaluation and optimisation in inductive databases using version spaces. In *Proc. EDBT Workshop on DTDM*, 2002.

[4] L. De Raedt, M. Jaeger, S.D. Lee, H. Mannila. A theory of inductive querying. In *Proceedings of the 2nd IEEE Conference on Data Mining*, Mabaeshi, Japan, 2002, in press.

[5] T. Imielinski and H. Mannila: A database perspective on knowledge discovery. *Communications of the ACM*, 39(11):58–64, 1996.

[6] S. Kramer and L. De Raedt: Feature construction with version spaces for biochemical applications. *Proceedings of the 18th International Conference on Machine Learning*, 258–265, Morgan Kaufmann, 2001.

[7] S. Kramer, L. De Raedt, C. Helma. Molecular Feature Mining in HIV Data. In *Proc. SIGKDD*, 2001.

[8] C. Mellish. The description identification algorithm. *Artificial Intelligence*, Vol. 52 (2), pp,. 151-168, 1990.

[9] A. Inokuchi, T. Washio, H. Motoda. An Apriori-based algorithm for mining frequent substructures from graph data. in D. Zighed, J. Komorowski, and J. Zyktow (Eds.) *Proceedings of PKDD 2000*, Lecture Notes in Artificial Intelligence, Vol. 1910, Springer-Verlag, 2000.

Computational Scientific Discovery and Inductive Databases

Sašo Džeroski
Department of Intelligent Systems
Jožef Stefan Institute
Jamova 39, SI-1000 Ljubljana, Slovenia

Computational scientific discovery (Langley et al. 1987; Shrager and Langley 1990) is concerned with applying computational methods to automate scientific activities. Early research on computational discovery (Langley et al. 1987) focussed on reconstructing episodes from the history of science by modeling the scientific activities and processes that led to the scientist's insight. Recent efforts in this area (for overviews see Langley 2000; Džeroski and Todorovski 2003) have focussed on individual scientific activities (such as formulating quantitative laws). Much of the work in computational scientific discovery has put emphasis on formalisms used to communicate among scientists, including numeric equations, structural models, and reaction pathways.

Over the last decade, we have developed a number of approaches to discovering quantitative laws in the form of equations or equation discovery. We have considered algebraic, ordinary differential (ODEs) and partial differential (PDEs) equations (Džeroski and Todorovski 1993; Todorovski and Džeroski 1997; Todorovski et al. 2000). The approaches developed have been applied to a number of practical modeling problems, mainly in the area of ecology (Todorovski et al. 1998; Džeroski et al. 1999). We have devoted special attention to the use of various forms of domain knowledge: we use declarative bias (Todorovski and Džeroski 1997) and background knowledge (Džeroski and Todorovski 2001), and also address the problem of revising theories that consist of quantitative laws (Todorovski and Džeroski 2001). The use of context-free grammars to define the space of equations considered (Todorovski and Džeroski 1997) allows us to treat all three forms of domain knowledge in a uniform way.

Using domain knowledge allows for a realistic approach to learning in difficult domains. Rather than trying to solve a difficult problem by starting from scratch, one can use existing domain knowledge in addition to collected observations (examples) and build upon it. Different types of domain knowledge can be taken into account, such as concepts already in common use (background knowledge), intuitions about the form of the target theory (declarative bias) and existing theories (theory revision). In this context, one can trade-off between the quantity and quality of observations and domain knowledge: high quantities of quality data may suffice to generate a good theory even with no domain knowledge, while smaller quantities of (lower quality) data may suffice if relevant domain knowledge is available.

Inductive databases (Imielinski and Mannila 1996) embody a database perspective on knowledge discovery, where knowledge discovery processes are considered as query processes. In addition to normal data, inductive databases contain patterns (either materialized or defined as views). Data mining operations looking for patterns are viewed as queries posed to the inductive database. In addition to patterns (which are of local nature), models (which are of global nature) can also be considered.

A general formulation of data mining (Mannila and Toivonen 1997) involves the specification of a language of patterns and a set of constraints that a pattern has to satisfy with respect to a given database. The constraints that a pattern has to satisfy can be divided in two parts: language constraints and evaluation constraints. The first only concern the pattern itself, the second concern the validity of the pattern with respect to a database. Constraints thus play a central role in data mining and constraint-based data mining is now a recognized research topic (Bayardo 2002).

Different types of patterns have been considered in data mining, including frequent itemsets, episodes, Datalog queries, and graphs. Designing inductive databases for these types of patterns involves the design of inductive query languages and solvers for the queries in these languages. For each type of pattern, or pattern domain, a specific solver is designed, following the philosophy of constraint logic programming (De Raedt 2002).

To bring equation discovery and inductive databases together, we consider inductive databases on the pattern domain of equations. When designing a query language for a given pattern domain, the language and evaluation constraints that are to be considered need to be specified. Language-wise, one might consider polynomial equations and search for sub-polynomials of a given polynomial

which have a high correlation coefficient with a dependent variable on the data at hand. Other evaluation measures can be considered, such as maximum absolute error, mean squared error, etc. Similarity language constraints can also be used: one can search for equations that are as similar as possible to a given equation and have a correlation coefficient above a certain threshold on a given dataset. The latter is essentially a theory revision problem in equation discovery (Todorovski and Džeroski 2001).

Inductive databases and constraint-based data mining open the door to more intensive use of domain knowledge in data mining and thus bring it closer to computational scientific discovery. In the pattern domain of equations, inductive queries that would allow for a combination of data-driven modeling and modeling from first principles would be possible. These would include queries that perform theory revision on models consisting of sets of equations.

In a given application domain, an inductive database would contain not only data about the domain but also models or model components (patterns). Alternative models of different aspects of the domain can be stored. Inductive queries would generate models from data only, from the data and model components, or revise models in light of the data. Computational scientific discovery can then be supported through inductive query sessions, which allows for a much more active role of the user as compared to traditional data mining.

Note that this calls for more effort on documenting and storing the results of the modeling process. Within he inductive database paradigm, this could provide strong support for further modeling activities (computational scientific discovery). While it is usual to have datasets in databases, models are typically not stored in databases. Efforts to create databases of models in different fields of science, such as the ECOBAS initiative in the field of ecological modeling, should thus be encouraged and supported. Hopefully this would facilitate computational scientific discovery by synthesizing new data and existing knowledge into new knowledge through the interaction of scientists with inductive databases.

References

[1] Bayardo, R., editor (2002). Constraint-based data mining. Special issue. *SIGKDD Explorations*.

[2] De Raedt, L. (2002). Data mining as constraint logic programming. In *Computational Logic: From Logic Programming into the Future (In honor of Bob Kowalski)*. Springer, Berlin.

[3] Džeroski, S., & Todorovski, L., editors (2003). *Computational Discovery of Communicable Knowledge*. Springer, Berlin. Forthcoming.

[4] Džeroski, S., & Todorovski, L. (2002). Encoding and using domain knowledge on population dynamics in equation discovery. In L. Magnani, N. J. Nersessian, and C. Pizzi, (editors), *Logical and Computational Aspects of Model-Based Reasoning*. Kluwer, Dordrecht.

[5] Džeroski, S., Todorovski, L., Bratko, I., Kompare, B., & Križman, V. (1999). Equation discovery with ecological applications. In A.H. Fielding, editor, *Machine Learning Methods for Ecological Applications* (pp. 185–207). Kluwer, Dordrecht.

[6] Džeroski, S., & Todorovski, L. (1993). Discovering dynamics. In *Proc. 10th International Conference on Machine Learning* (pp. 97–103). Morgan Kaufmann, San Mateo, CA.

[7] Imielinski, T., and Mannila, H. (1996). A database perspective on knowledge discovery. *Communications of the ACM*, 39(11): 58–64.

[8] Langley, P. (2000). The computational support of scientific discovery. *International Journal of Human-Computer Studies*, 53: 393–410.

[9] Langley, P., Simon, H.A., Bradshaw, G.L., & Zytkow, J. (1987). *Scientific Discovery*. MIT Press, Cambridge, MA.

[10] Mannila, H., and Toivonen, H. (1997). Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3): 241–258.

[11] Shrager, J., & Langley, P., editors (1990). *Computational Models of Scientific Discovery and Theory Formation*. Morgan Kaufmann, San Mateo, CA.

[12] Todorovski, L., & Džeroski, S. (2001). Theory revision in equation discovery. In *Proc. 4th International Conference on Discovery Science* (pp. 390–400). Springer, Berlin.

[13] Todorovski, L., Džeroski, S., Srinivasan, A., Whiteley, J., & Gavaghan, D. (2000). Discovering the structure of partial differential equations from example behavior. In *Proc. 17th International Conference on Machine Learning* (pp. 991–998). Morgan Kaufmann, San Francisco, CA.

[14] Todorovski, L., Džeroski, S., & Kompare, B. (1998). Modeling and prediction of phytoplankton growth with equation discovery. *Ecological Modelling* 113: 71–81.

[15] Todorovski, L., & Džeroski, S. (1997). Declarative bias in equation discovery. In *Proc. 14th International Conference on Machine Learning* (pp. 376–384). Morgan Kaufmann, San Francisco, CA.

Active Learning Approaches For Scaling Up Discovery Algorithms

Stefan Wrobel

Fraunhofer AiS, Schloß Birlinghoven 53754 Sankt Augustin, Germany
E-Mail wrobel@ais.fhg.de
and
University of Bonn, Informatik III, Römerstr. 164, 53117 Bonn, Germany
E-Mail wrobel@cs.uni-bonn.de

Extended Abstract

The typical machine learning or knowledge discovery algorithm takes as input a database of observations, facts or examples and processes this database in its entirety to learn the optimal model or discover the set of most interesting hypotheses. With respect to the amount of data used by such an algorithm, we call such algorithms *passive* in the following sense: The algorithm is optimized to using whatever data it is presented with, and makes no attempt to evaluate the appropriateness of the data or select the most useful part. In contrast, an *active* learning algorithm will not simply perform its search using all the given data, but will try to actively select from the given data the amount and type of data that are required or useful for reaching its specified learning goal. While typically the quality of results is no different when learning actively or passively, the required resources can differ by several orders of magnitude.

In this talk, we introduce the general idea of active learning, and then provide more detail on three different scenarios of active learning, namely selection of the *right amount* of data, selection of the *right data points* when given unlabeled data, and selection of the *right features and relations* in a propositional or multirelational learning setting. From our own work, we will show realizations of these three scenarios using different learning tasks (deviation detection/subgroup discovery, learning of Hidden Markov models, and classical predictive learning).

Active selection of the amount of data

In knowledge discovery, we are typically given very large databases, often too large to be processed in their entirety. In this situation, actively deciding how much of the data needs to be looked at to reach a given learning goal can make the difference between being able to handle a given learning problem or not being able to handle it at all. In this setting, active learning thus amounts to a principled approach to *sampling*, where the sample size is chosen just high enough for the desired quality of learning results. In the past years[1], we have been examining active learning algorithms of this type for the task of discovering the n best hypotheses in a database as defined by a given utility or quality function (joint work with T. Scheffer, published as [7, 8, 9, 5]).

Definition 1 (Approximate n-best hypotheses problem) *Let D be a distribution on instances, H a set of hypotheses, $f : H \rightarrow \mathbb{R}^{\geq 0}$ a function that assigns a utility value to each hypothesis and n a number of desired solutions. Then let δ, $0 < \delta \leq 1$, be a user-specified confidence, and $\varepsilon \in \mathbb{R}^+$ a user-specified maximal error. The* approximate n-best hypotheses problem *is to find a set $G \subseteq H$ of size n such that*

> *with confidence $1 - \delta$, there is no $h' \in H$: $h' \notin G$ and $f(h', D) > f_{min} + \varepsilon$, where $f_{min} := min_{h \in G} f(h, D)$.*

In order to solve this problem efficiently, the use of active techniques is essential. In a static approach to sampling, one would use standard statistical tools to derive confidence intervals for certain sample sizes. In order to make sure that we find the truly best hypotheses, we then need to choose a sample size that limits the confidence interval of each hypothesis to $\frac{\varepsilon}{2}$, since this guarantees that the apparent ordering of our hypotheses is indeed the correct one. As we empirically demonstrated in [8], this approach results in impractically large sample sizes.

Instead, one must resort to an active approach based on *sequential sampling*. In such an approach, the estimates and confidence intervals of all hypotheses are constantly monitored while new samples are being looked at. As soon as we can be certain that a hypothesis must be a solution, we

[1] Parts of the work mentioned in this extended abstract were carried out while the author was still at University of Magdeburg.

can already output it. As soon as we know that a hypothesis cannot be a solution, it can already be removed. The detailed algorithm is described in [8, 5] and is proved correct in the latter paper. More importantly, empirical experiments on artificial and on real-world data show that this approach results in a speed up that can be as high as four orders of magnitude, i.e., a factor of 10.000. With a recent extension of the algorithm [9], it is possible to follow this active approach to sampling even when only a small constant memory space is available.

Active selection of the right data points

In many situations, we do not necessarily have access to a large set of data points labeled with the desired target value, but can readily access large collections of unlabeled objects. This is the case for example in any application where the goal is to learn from text. Here, we usually must rely on the user to provide labels for documents or individual tokens, which is an expensive process, while at the same time, we can simply access the Internet or a user specific corpus to obtain basically as many documents as we would like. In this situation, active learning can go beyond deciding how many data point to look at, and can decide *which* of the yet unlabeled data points to look at, i.e., for which data points we want to ask for the (expensive to obtain) label. If done well, looking at an additional n data points selected actively should allow a better learning result than looking at n randomly selected data points. In our own work, we have applied this idea to the problem of learning Hidden Markov models (HMMs) for information extraction and document classification (joint work with T. Scheffer, B. Popov, D. Ognianov, C. Decomain and S. Hoche, published as [4, 3, 6]).

To this end, we have developed an algorithm for learning HMMs on textual documents that are only partially labeled. Just as the standard algorithm for learning Hidden Markov models, our algorithm uses the iterative EM technique for identifying the hidden states of the HMM, but uses update rules which handle unlabeled tokens in the correct fashion. The EM nature of the algorithm allows us to perform active learning in the following way. When trying to determine the hidden state corresponding to the token in the document, the algorithm produces a probability distribution across the set of possible states (and thus labels). One can then use the simple idea of a *margin* to select tokens the label of which is to be requested. We define the margin as the difference in probability between the winning state and the next best alternative. If this difference is small, we know that a few additional data points could have changed the ordering of states, and thus would lead us to selecting a different state. A token with a small margin is thus classified with low certainty, so that we should actively ask about the labels for tokens with the smallest margins in order to increase our certainty. Indeed it turns out that this active learning strategy is capable of reducing the amount of labeled data that is required: in our experiments, we were able to reach the same level of error of the learned HMM with as little as one fifth of the amount of data necessary when choosing randomly.

Active selection of the right features and relations

Active learning can go beyond selecting data points from a given table of data. Efficiency and effectiveness also depend on the *representation* of the objects that are used for learning. The inclusion of unnecessary features or, in a multirelational setting, of unnecessary relations, makes learning less efficient and often less effective. When considering learning in a large data warehouse with dozens of relations with dozens of attributes each — a size not uncommon in commercial data warehouses — being able to perform this selection is absolutely essential. While it will probably always be true that such a data engineering process is best performed by a human with domain knowledge, it seems that active learning offers encouraging potential as well. In our own work, we have examined this question for propositional and multirelational settings based on a boosted learning algorithm (joint work with S. Hoche, published as [1, 2]).

Boosting, an *ensemble* method for learning, uses an iterative process to produce a learning result that consists of a set of hypotheses which are combined using a voting process to actually arrive at a prediction. In each iteration of boosting, a base learner is called on the data to produce one hypothesis for the final set. In the first iteration, the weight of each example of the set is identical. From then on, the distribution of weights changes with each iteration, i.e., the base learner must be capable of handling weight information with examples. For the next iteration, the weight of an example that was classified correctly by the hypothesis produced in the preceding iteration is reduced, while the weight of an example that was classified incorrectly is increased. In this fashion, boosting concentrates more and more on the examples which are difficult to classified correctly. For our work, we have used a particularly restricted form of boosting combined with a simple and efficient multirelational base learner [1].

In order to perform active selection of features and relations here, one can use the very properties of boosting. Since boosting tries to increase the certainty with which examples are classified by the ensemble of hypotheses — this is expressed in terms of the so-called *margin* —, one can monitor the development of the margin in order to see when new features or relations might be needed. To this end, one orders the available features in the different relations based on a heuristic relevance measure. Boosting is then started

with a minimal set of features, and proceeds until the development of the margin indicates that progress is slowing down at which moment the next feature on the list is added to the representation. In empirical experiments, we have found that this approach can reduce learning times quite significantly, sometimes even by an order of magnitude, while maintaining or even increasing accuracy [2].

Towards Active Discovery

Clearly, the above three approaches are but simple examples of what active learning techniques could bring to knowledge discovery; there are already plenty of other approaches in the field that we have not mentioned here. In the long run, active learning could mean the construction of an autonomous discovery agent that is capable of reaching the discovery goals specified by the human user with a minimum of resources in terms of running time or money. Such an agent would have to resort to significantly more powerful mechanisms than described above, for example by constructing models of data sources or learning about the quality and usefulness of particular types of data for a given goal. In the end, this would relieve the human from all the tedious tasks of data preprocessing and data engineering that constitute most of a data mining project today — but it is probably best to abstain from an estimate of when such a goal can indeed be reached.

Acknowledgements

The research reported here was partially supported by Grant "Information Fusion / Active Learning" of the German Research Council (DFG).

References

[1] S. Hoche and S. Wrobel. Relational learning using constrained confidence-rated boosting. In *Proc. 11th Int. Conference on Inductive Logic Programming*, pages 51 – 64. Springer-Verlag, 2001.

[2] S. Hoche and S. Wrobel. Scaling boosting by margin-based inclusion of features and relations. In *Proc. 13th European Conference on Machine Learning*, pages 148 – 160. Springer-Verlag, 2002.

[3] T. Scheffer, C. Decomain, and S. Wrobel. Active hidden markov models for information extraction. In *Advances in Intelligent Data Analysis, 4th International Conference, IDA 2001*, pages 309–318. Springer Verlag, 2001.

[4] T. Scheffer, C. Decomain, and S. Wrobel. Mining the web with active hidden markov models. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 645–646. IEEE Computer Society, 2001.

[5] T. Scheffer and S. Wrobel. Finding the most interesting patterns in a database quickly by using sequential sampling. *Journal of Machine Learning Research*, to appear.

[6] T. Scheffer, S. Wrobel, B. Popov, D. Ognianov, C. Decomain, and S. Hoche. Learning hidden markov models for information extraction actively from partially labeled text. *Künstliche Intelligenz*, 2002.

[7] T. S. und Stefan Wrobel. A sequential sampling algorithm for a general class of utility criteria. In *Proc. 6th International Conference On Knowledge Discovery and Data Mining*, pages 330 – 334. ACM Press, 2000.

[8] T. S. und Stefan Wrobel. Incremental maximization of non-instance-averaging utility functions with applications to knowledge discovery problems. In *Proc. 12th International Conference On Machine Learning*, pages 481–488. Morgan Kaufman, 2001.

[9] T. S. und Stefan Wrobel. A scalable constant-memory sampling algorithm for pattern discovery in large databases. In *Principles of Data Mining and Knowledge Discovery, 6th European Conference, PKDD 2002*, pages 397–409, 2002.

Relational Ranking with Predictive Clustering Trees

Sašo Džeroski and Ljupčo Todorovski
Department of Intelligent Systems
Jožef Stefan Institute
Jamova 39, SI-1000 Ljubljana, Slovenia

Hendrik Blockeel
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A, B-3001 Heverlee, Belgium

Abstract

A novel class of applications of predictive clustering trees is addressed, namely relational ranking. Predictive clustering trees, as implemented in TILDE, allow for predicting multiple target variables from relational data. This approach makes sense especially if the target variables are not independent of each other. This is typically the case in ranking, where the (relative) performance of several approaches on the same task has to be predicted from a given description of the task.

We propose to use predictive clustering trees for ranking. This allows us to use relational descriptions of the tasks. As compared to existing ranking approaches which are instance-based, our approach also allows for an explanation of the predicted rankings. We illustrate our approach on the task of ranking machine learning algorithms, where the (relative) performance of the algorithms on a given dataset has to be predicted from a given (relational) dataset description.

1. Introduction

In many cases, running an algorithm on a given task can be time consuming, especially when complex tasks are involved. It is therefore desirable to be able to predict the performance of a given algorithm on a given task from a description (set of properties of the task) and without actually running the algorithm. The term "performance of an algorithm" is often used to denote the quality of the solution provided, the running time of the algorithm or some combination of the two.

As an example, consider the task of optimization, e.g., finding the minimum value of a function. Given an optimization algorithm (say Levenberg-Marquardt), we might be interested in predicting the quality of the solution found (e.g., how close to the real optimum was the solution) and/or the running time of the algorithm. A description of the task would be a description of the function to be optimized (e.g., in terms of the number of different trigonometric and algebraic operators appearing in it, the size of the tree needed to encode the function, etc.).

When several algorithms are available to solve the same type of task, the problem of choosing an appropriate algorithm for the particular task at hand arises. We can view this as a multi-target prediction problem, where the same input (the task description) is used to predict several related targets (the performances of the different algorithms). In this context, it is the relative performance of the different algorithms that matters, and not so much the absolute performance of each of them. We are thus interested in obtaining an ordering of the algorithms (called ranking) in terms of their expected relative performance.

Within the area of machine learning, many learning algorithms have been developed, especially for classification tasks. A classification task is specified by giving a table of data and indicating the target column: the pair is often referred to as a dataset. The task of predicting the performance of learning algorithms from dataset properties has been addressed within the StatLog project [5], while the task of ranking learning algorithms has been one of the major topics of study of the METAL project [1]. Both are treated as learning problems, where the results of applying selected learning algorithms on selected datasets (base-level learning) constitute a dataset for meta-level learning.

A typical meta-level dataset for ranking thus consists of two parts. The first set of columns (attributes) contain a description of the task at hand. In the case of ranking learning algorithms, it typically contains general and statistical properties of datasets (such as the number of examples and class values and the average kurtosis per numerical attribute). The second set of columns contains the performance figures for the learning algorithms on the given datasets (e.g., the classification error of C5.0, RIPPER, etc.).

Many different variants of ranking have been studied within the METAL project. A prototypical ranker uses a case-based (nearest neighbor) approach. To produce a rank-

ing on the learning algorithms a new dataset, the most similar datasets from the meta-level dataset are chosen and the performances (rankings) of the algorithms on theses datasets are averaged to obtain a prediction of the performance (ranking) on the new dataset.

In this paper, we propose to use a relational representation of tasks (datasets) in ranking instead of a propositional one. This allows us to represent the tasks in more detail, e.g., include the kurtosis values for each numerical attribute rather than include only the average kurtosis per numerical attribute. We also propose to use predictive clustering trees for ranking instead of case-based approaches. In this case, in addition to obtaining a ranking, we also obtain an explanation.

The remainder of this paper is organized as follows. Section 2 describes in more detail the task of relational ranking of learning algorithms. This includes the base-level datasets, the algorithms ranked, the performance evaluation methodology, and finally the propositional and relational descriptions of datasets. Section 3 briefly describes predictive clustering trees and describes the particular formulation of the multi-target (relative) performance prediction used in our experiments. Section 4 describes the experimental setup and the results of evaluating our approach to ranking learning algorithms. Finally, Section 5 concludes with a summary and possible directions for future work.

Table 1. Ten machine learning algorithms for classification tasks used in our study.

Acronym	Brief description
c50tree	C5.0 - decision trees based classifier
c50rules	decision rules extracted from a C5.0 tree
c50boost	boosting C5.0 decision trees
ltree	linear discriminant decision trees
ripper	decision rules based classifier
mlcnb	naive Bayes classifier (MLC++)
mlcib1	1-NN nearest neighbor classifier (MLC++)
lindiscr	linear discriminant classifier
clemMLP	multilayer perceptron ANN (Clementine)
clemRBFN	radial-basis functions ANN (Clementine)

2. Relational Ranking of Learning Algorithms

This section describes in more detail the task of relational ranking of learning algorithms. This includes the algorithms ranked, the base-level datasets, the propositional and relational descriptions of the datasets, and the performance evaluation methodology.

2.1 The machine learning algorithms

In this study, we analyze the relative performance of ten machine learning algorithms for classification tasks. The list of algorithms is presented in Table 1: these are the ten algorithms used within the METAL project [1]. This set includes one or more representatives of different classification approaches, such as decision trees and rules, naive Bayes, nearest neighbor and linear discriminant classifiers, as well as neural networks.

Table 2. Forty-two classification datasets used in our study.

abalone	adult	allbp
allhyper	allhypo	allrep
ann	byzantine	c_class_flares
car	contraceptive	dis
dna_splice	fluid	german_numb
german_symb	krkopt	letter
m_class_flares	mushrooms	musk
nettalk	nursery	optical
page	pendigits	pyrimidines
quadruped	quisclas	segment
shuttle	sick	sick_euthyroid
splice	taska_part_hhold	taska_part_related
taskb_hhold	triazines	waveform21
waveform40	x_class_flares	yeast

2.2 The datasets

The performance of these ten algorithms have been measured on a set of forty-two classification tasks (datasets) used within the METAL project.[1] The list of datasets is given in Table 2. Some of these come from the UCI Repository of Machine Learning Datasets, while others are proprietary.

2.3 Dataset descriptions

Each classification task from Table 2 is described using a set of task properties. In the StatLog project, a set of general, statistical and information theory based dataset properties has been used [5] for dataset description. This gave rise to the Data set Characterizing Tool (DCT) [7], developed further within the METAL project [8], that extends

[1] Fifty-three classification tasks are considered within the METAL meta-level learning studies. However, in our study we have used only a subset of forty-two classification tasks, where the meta-level data (both properties and performance measures) were available. We will include the whole set of METAL classification tasks in our study, as soon as meta-level data become available.

Table 3. Data set properties.

Whole dataset	
num_of_attrs	num_of_sym_attrs
num_of_num_attrs	num_of_examples
num_of_classes	missing_values
lines_with_missing_values	mean_skewness
mean_kurtosis	num_of_attrs_with_outliers
M_stat	M_stat_DF
M_stat_ChiSq	M_stat_ChiSq_alpha
SD_ratio	fract
cancor	wilks_lambda
Bartlett_stat	Bartlett_stat_DF
Bartlett_stat_ChiSq	Bartlett_stat_ChiSq_alpha
class_entropy	entropy_attributes
joint_entropy	equivalent_num_of_attrs
noise_signal_ratio	perc_sym_attrs
perc_num_attrs	examples_per_attr
classes_per_attr	rel_num_of_attrs_with_outliers
rel_equivalent_num_of_attrs	log_num_of_examples

Per attribute	Aggregates
perc_missing_values	AVG MIN MAX
skewness	AVG MIN MAX
kurtosis	AVG MIN MAX
multi_correl	AVG MIN MAX
gini_index	AVG MIN MAX
relevance	AVG MIN MAX
g_function	AVG MIN MAX
class_freq	MAX

the set of StatLog properties. We included most of the DCT properties in the dataset descriptions used in this study. The complete list of dataset properties is presented in Table 3.

There are two groups of DCT properties. The first group contains properties of the entire dataset (first column in Table 3), while the second group contains properties of individual attributes in the dataset (second column in Table 3). In addition, the probability distribution of the class is also included.

The general DCT properties include simple facts about the dataset, such as number of examples, (nominal and numeric) attributes and class values, but also more complicated statistical and information theory based measures of the whole dataset. Furthermore, six measures are used to characterize individual attributes. Three of them are statistical measures for numerical attributes and three of them are information theory based measures for discrete attributes.

Properties of the individual attributes can not be used directly in propositional meta-learning, where the dataset description is a fixed-length vector of dataset properties. For this purpose, each property of the individual attributes is aggregated using the average, minimum or maximum function. The relational framework for meta-learning allows for a more complex representation of data sets [11]. In this study, we include all the DCT properties from Table 3, both global properties of the entire dataset (the general and aggregated ones) and properties of individual attributes.

2.4 The performance of a learning algorithm

When building a dataset for meta-learning, we also need an estimate of the performance of the learning algorithms on a given classification task. Most often, the performance of a learning algorithm a on a given classification task d is measured by the predictive accuracy $ACC(a, d)$, i.e., the percentage of correctly classified examples. To estimate this predictive accuracy on test examples, unseen during the training of the classifier, a standard ten-fold cross validation method has been used. Another performance measure of a learning algorithm a is its running time $T(a, d)$ on a given dataset d. A third performance measure that combines the predictive accuracy with the running time of a machine learning algorithm named "*adjusted ratio of ratios*" has been proposed in [10]:

$$ARR(a_p, d) = \sum_{a_q \in A, a_q \neq a_p} ARR(a_p, a_q, d),$$

$$ARR(a_p, a_q, d) = \frac{\frac{ACC(a_p,d)}{ACC(a_q,d)}}{1 + \frac{\log\left(\frac{T(a_p,d)}{T(a_q,d)}\right)}{K_T}}$$

where A is the set of learning algorithms under study, and K_T is a user-defined value that determines the relative importance of the running time. The K_T parameter is approximated by $K_T = 1/X\%$, where X is the accuracy one is willing to trade for a 10 times speedup or slowdown. However, due to the lack of the data about running time of the algorithms, we used the ARR measure with the setting $K_T = \inf$ which eliminates the influence of time.

3. Relational Ranking with Predictive Clustering Trees

This section first briefly describes predictive clustering trees. It then discusses how they could be used to predict the accuracies of different learning algorithm on a given dataset simultaneously. It finally proposes to use the ranks calculated from the accuracies as the target variables, rather than the accuracies themselves.

3.1 Predictive Clustering Trees

A variety of algorithms for predictive modeling exists. Among the better known are algorithms that induce decision trees [6, 9]. Compared to other well-known techniques

such as neural networks [2], decision trees have the advantage of being more interpretable: they clearly explicitate the factors that influence the outcome most strongly.

Decision trees are most often used in the context of classification or single-target regression; i.e., they represent a model in which the value of a single variable is predicted. However, as a decision tree naturally identifies partitions of the data (course-grained at the top of the tree, fine-grained at the bottom), one can also consider a tree as a hierarchy of clusters. A good cluster hierarchy is one in which individuals that are in the same cluster are also similar with respect to a number of observable properties.

This leads to a simple method for building trees that allow the prediction of multiple target attributes at once. If we can define a distance measure on tuples of target variable values, we can build decision trees for multi-target prediction. The standard TDIDT algorithm can be used: as a heuristic for selecting tests to include in the tree, we use the minimization of intra-cluster variance (and maximization of inter-cluster variance) in the created clustering.

A detailed description of the algorithm (called TIC) can be found in [3]. We used the implementation of TIC as available in the first-order learner TILDE that is included in the ACE tool [4]. This implementation allows for relational tests to be used in the nodes of predictive clustering trees through the use of declarative bias.

3.2 Ranking via Predicting Errors

The instance-based approaches to ranking predict rankings of algorithms on a dataset by predicting the accuracies of the algorithms on the dataset, then creating a ranking from these. An instance here consists of a description of a dataset, plus the performance of 10 different algorithms on that dataset (this performance can be measured as accuracies or ARR values). Based on these 10 target values, an example can be positioned in a 10-dimensional space.

In its standard mode of operation, TILDE builds its trees so that the intra-cluster variance is minimized, where variance is defined as

$$\sum_{j=1}^{N} d(\mathbf{x}_j, \bar{\mathbf{x}})^2$$

where $\bar{\mathbf{x}}$ is the mean vector of the cluster, $\mathbf{x}_j$ is an element of the cluster, N is the number of elements in the cluster, and d represents the euclidean distance. So, what TILDE does is trying to create clusters in such a way that a given algorithm will perform similarly on all datasets in that cluster.

Note that this is different from what we want: creating clusters in which several algorithms have the same relative performance. To illustrate this, suppose we have 4 algorithms which on 2 datasets score the following accuracies:

$$\{(0.1, 0.2, 0.3, 0.4), (0.5, 0.6, 0.7, 0.8)\}$$

Clearly the relative performance of the 4 algorithms is exactly the same on the three datasets, so they belong to the same cluster. However, the variance in this cluster is relatively large. Compare this to

$$\{(0.1, 0.2, 0.3, 0.4), (0.4, 0.3, 0.2, 0.1)\}$$

which has a smaller variance than the previous cluster but is clearly worse: the relative performances are opposite.

3.3 Ranking Trees

A solution for this problem is to first rank the algorithms and to predict these ranks instead of the accuracies themselves. In this way, we obtain ranking trees. A ranking tree has leaves in which a ranking of the performance of different algorithms is predicted.

This transformation removes fluctuations in the variance that are caused by differences in absolute rather than relative performance. Moreover, given the formula for the Spearman correlation:

$$r_s = 1 - 6\frac{(\sum_{i=1}^{n} {D_i}^2)}{n^3 - n}$$

where D_i is the difference between actual and predicted rank of the i'th algorithm and n is the number of learning algorithms, it is clear that a linear relationship between variance and expected Spearman correlation exists. Indeed, note that

$$d(\mathbf{x}, \bar{\mathbf{x}})^2 = \sum_{i=1}^{n} {D_i}^2$$

on the condition that the "predicted rank" in each leaf of the tree is indeed the number found for the algorithm.

The latter condition is a problem. The predictive clustering tree might predict, in a specific case,

$$(6.7, 6.0, 6.4, 3.65, 6.1, 5.65, 3.5, 5.65, 3.7, 7.65)$$

If we would use these numbers as predictions, minimizing intra-cluster variance would be equivalent to maximizing expected correlation. However, if we rank algorithms based on these numbers, i.e. use the ranks of the numbers

$$(9, 6, 8, 2, 7, 4.5, 1, 4.5, 3, 10)$$

instead of the original numbers, then the equivalence does not hold anymore, and minimizing intra-cluster variance should be seen as an approximation to maximizing Spearman correlation.

4. Experiments

Our experiments investigate the performance of relational ranking with predictive clustering trees on the dataset

described in Section 2. Following the discussion from Section 3.3, we transformed the target accuracies and ARR values into ranks. Doing this, we noticed that both performance measures result into the same ranking of the learning algorithms, thus from now on, we do not make distinction between these two performance measures. The remainder of this section first describes the experimental setup, and in particular the language biases used, and the pruning performed. It then presents the experimental results, including an example ranking tree and performance figures on the correlation between actual and predicted rankings.

4.1 Experimental Setup

The TILDE system has a number of settings that influence its behavior. We have performed several experiments, using default values for all settings except the following settings, which were varied (an explanation follows):

- Language bias: None, Prop, Rel, Both
- Ftest settings: 0.001, 0.005, 0.01, 0.05, 0.1, 1.0

The four language bias settings correspond to using only propositional information (i.e., properties of the whole dataset and aggregations of the properties of the individual attributes/classes), only relational information (i.e, properties of individual attributes/classes only), both, or no information at all. The latter bias was included to measure the performance of "default" models that consists of just a leaf (these predict the average of the values encountered in the training set). For the propositional data, the language bias consists of tests $A < c$ with A a meta-attribute (all meta-attributes are numeric) and c some value for it (any value from A's domain was allowed). The number of A, c combinations (i.e., the number of possible tests that can be constructed at a single node) is over 1400.

For the relational data, a language was constructed that essentially allows to check properties of an individual attribute of a dataset, e.g., check if the skewness of some numeric attribute of that dataset is above a certain value. Note that checking for the existence of an attribute with skewness > 1 (for instance) is equivalent to checking whether the maximum of all skewness values is > 1. As such maxima (and minima) are included in the propositional descriptions of the datasets, this in itself does not yield more expressiveness. With the relational version it is however also possible to check for the existence of a single attribute in a dataset that has several properties, e.g., "is there an attribute with skewness > 1 and kurtosis < 0"; this kind of tests cannot be constructed in our propositional representation. With the relational representation, the number of tests considered at a specific node of the tree varies from 566 to over 1000.

The Ftest setting in TILDE is a stopping heuristic based on the classical statistical F-test. The values indicate significance levels; lower values cause the tree to be smaller. We have not exhaustively searched the space of all possible parameter settings, but instead explored it more or less intuitively, in a kind of hill-climbing fashion. More specifically, we first performed the following experiment: "With language Rel and varying Ftest from 0.001 to 1.0, estimate the performance of ranking trees using leave-one-out". The results suggested that best performance is obtained at high Ftest values, i.e., with the least-pruned trees.

Table 4. Performances of PCTs induced using three different biases (propositional, relational and both) compared to the default performance (of a single-node PCT).

	$F = 0.05$		$F = 0.1$		$F = 1.0$	
	RE	r_s	*RE*	r_s	*RE*	r_s
propositional	1.16	0.46	1.13	0.49	1.20	0.46
relational	1.06	0.47	0.94	0.49	0.87	0.53
both	1.12	0.47	1.10	0.49	1.16	0.48
default	1.0	0.51	1.0	0.51	1.0	0.51

4.2 Experimental Results

We next performed a second experiment: "For all language biases, and for large Ftest values (0.05, 0.1, 1.0), estimate the performance of ranking trees using leave-one-out." The results are reported in Table 4. RE is the relative error as estimated by TILDE using leave-one-out. r_s is the average Spearman rank correlation between the predicted ranking and the actual ranking of a left-out instance. The table makes clear that the best results are obtained for relational data with an F-test value of 1.0: here the RE is lowest, and the r_s is highest.

There are a few interesting observations to make.

- Results in general are not very good, with most RE's over 1.0 (i.e. worse than default prediction, squared-error-wise) and, unsurprisingly in this light, most r_s below the default r_s of 0.51. It is somewhat strange that a learner would almost consistently construct theories worse than default; we suspect that the very small datasets and the pessimistic bias of cross-validation causes these results to look somewhat worse than they really are.
- The best result is obtained for the relational representation. A closer look at the induced trees reveals that

Table 5. An example ranking tree. In each leaf node the ranking of ten machinge larning algorithms is predicted. The following labels are used to dentote learning algorithms (See also Table 1): c5t = c50tree, c5r = c50rules, c5b = c50boost, lt = ltree, rip = ripper, nb = mlcnb, ib = mlcib1, ld = lindiscr, mlp = clemMLP, rbfn = clemRBFN. The '<' sign is used to dentoe the relation 'performs worse than'.

```
err_ranks(A,B,C,D,E,F,G,H,I,J,K)
classvalue_freq(A,L,M),M < 0.165 ?
+--yes:attr_skew_all(A,N,O),O>3.64 ?
|     +--yes:classvalue_freq(A,P,Q),Q>0.318 ?
|     |     +--yes:classvalue_freq(A,L,R),R < 0.097 ?
|     |     |     +--yes:attr_gfunction(A,S,T),T> -0.437 ?
|     |     |     |     +--yes:attr_relevance(A,S,U),safe(U>0.235) ?
|     |     |     |     |     +--yes:ld < nb < rbfn < ib < mlp < rip < lt < c5r < c5b < c5t
|     |     |     |     |     +--no: nb < ld < rbfn < mlp < lt < ib < rip < c5r < c5t < c5b
|     |     |     |     +--no: classvalue_freq(A,L,V),V < 0.003 ?
|     |     |     |            +--yes:rbfn < c5b < mlp < ld < nb < lt < ib < c5r < c5t < rip
|     |     |     |            +--no: rbfn < c5b < ld < nb < mlp < ib < rip < lt < c5t < c5r
|     |     |     +--no: mlp < rbfn < nb < ld < ib < rip < c5b < c5r < c5t < lt
|     |     +--no: attr_skew_all(A,N,W),W < 5.217 ?
|     |            +--yes:mlp < c5b < nb < rip < rbfn < c5t < ld < c5r < lt < ib
|     |            +--no: nb < ib < rbfn < mlp < c5t < lt < ld < c5r < rip < c5b
|     +--no: classvalue_freq(A,X,Y),Y>0.786 ?
|            +--yes:nb < ld < ib < c5b < rip < lt < mlp < c5r < c5t < rbfn
|            +--no: attr_skew_all(A,Z,A1),A1 <  -0.612 ?
|                   +--yes:ld < rip < c5t < nb < c5r < rbfn < lt < mlp < c5b < ib
|                   +--no: attr_relevance(A,B1,C1),safe(C1 < 0.062) ?
|                          +--yes:rbfn < ld < nb < rip < lt < c5t < mlp < c5r < ib < c5b
|                          +--no: c5b < ld < rbfn < nb < mlp < rip < lt < c5r < c5t < ib
+--no: attr_skew_all(A,D1,E1),E1>2.095 ?
      +--yes:attr_skew_all(A,F1,G1),G1 <  -0.26 ?
      |     +--yes:rbfn < mlp < nb < rip < ld < ib < c5b < c5r < c5t < lt
      |     +--no: nb < rbfn < ld < ib < c5t < lt < rip < c5r < mlp < c5b
      +--no: attr_skew_all(A,H1,I1),I1> -1.303 ?
             +--yes:attr_kurt_all(A,H1,J1),safe(J1 < 1.631) ?
             |     +--yes:ib < c5t < mlp < c5r < rip < rbfn < lt < c5b < nb < ld
             |     +--no: ib < c5t < rip < c5r < nb < c5b < rbfn < lt < ld < mlp
             +--no: mlp < rbfn < ib < ld < lt < c5r < c5t < rip < nb < c5b
```

the essentially relational aspects of the representation are used, but not very often, i.e., most of the trees could in principle also be found from propositional data. Our explanation for this is that the large number of attributes in the propositional descriptions confuses the tree learner. Somewhat unintuitively, the relational bias is actually stronger (there are fewer splits possible): consequently, the relational descriptions are more concise and can focus on more relevant properties.

We have taken a closer look at the tree in Figure 5, derived from the entire dataset with the optimal settings. The tree suggests, for instance, that the most important property of a dataset with respect to the behavior of learning algorithms is whether that dataset contains an infrequent class ("infrequent" defined here as having a frequency below 0.165). Also, the skewness of attribute distributions is identified as highly relevant. Finally, the tree uses relational information, for example, one leaf includes the condition: "there is an attribute with skewness above -1.303 and kurtosis below 1.631."

5. Summary and Further Work

We have used predictive clustering trees to rank (predict the relative performance of) machine learning algorithms for classification. A relational description of datasets is used, which allows to specify dataset properties in more detail: for example, properties of individual attributes can be used rather than bulk properties averaged across all attributes of a dataset. The relational ranking trees perform

better than propositional ranking trees and also better than the default ranking when a smaller amount of tree pruning is applied. As compared to existing ranking approaches which are propositional and instance-based, our approach also allows for an explanation of the predicted rankings.

An immediate direction for further work is to repeat the experimental evaluation on the full METAL ranking dataset (53 meta-level data points) once it becomes available. Given the size of the meta-level dataset, any additional point matters. This would also allow for a direct comparison to the propositional instance-based approaches to ranking.

Other directions for further work include the definition of a relational distance measure on datasets and the use of relational instance-based learning for relational ranking. Investigating the use of kernels for relational data would also be an interesting direction to pursue. Both approaches work well for small datasets of high dimensionality.

If we can deal with small datasets of high dimensionality, it makes sense to also consider additional dataset properties. Dataset properties based on landmarking have been shown to predict performance well and to be useful for ranking. Also, features based on the shape of decision trees induced from a datasets could be interesting in this respect.

Finally, the relational ranking methodology proposed in the paper can be also used and evaluated on other ranking tasks. A possible application would be the ranking of optimization algorithms on the basis of descriptions of optimization problems.

References

[1] *ESPRIT METAL Project (project number 26.357)*: A Meta-Learning Assistant for Providing User Support in Machine Learning and Data Mining. `http://www.metal-kdd.org/`.

[2] C. M. Bishop. *Neural Networks for Pattern Recognition*. University Press, Oxford, 1999.

[3] H. Blockeel, L. De Raedt, and J. Ramon. Top-down induction of clustering trees. In *Proceedings of the 15th International Conference on Machine Learning*, pages 55–63, 1998. `http://www.cs.kuleuven.ac.be/~ml/PS/ML98-56.ps`.

[4] H. Blockeel, L. Dehaspe, B. Demoen, G. Janssens, J. Ramon, and H. Vandecasteele. Improving the efficiency of inductive logic programming through the use of query packs. *Journal of Artificial Intelligence Research*, 16: 135–166, 2002.

[5] P. B. Brazdil and R. J. Henery. Analysis of results. In D. Michie, D. J. Spiegelhalter, and C. C. Taylor, editors, *Machine learning, neural and statistical classification*, pages 98–106. Ellis Horwood, 1994.

[6] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.

[7] R. Engels and C. Theusinger. Using a Data Metric for Offering Preprocessing Advice in Data Mining Applications. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence*, pages 430–434, 1998.

[8] G. Lindner and R. Studer. Ast: Support for algorithm selection with a cbr approach. In *Proceedings of the ICML-99 Workshop on Recent Advances in Meta-Learning and Future Work*, pages 38–47, 1999.

[9] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann series in machine learning. Morgan Kaufmann, 1993.

[10] C. Soares and P. B. Brazdil. Zoomed ranking: Selection of classification algorithms based on relevant performance information. In *Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery*, pages 126–135. Springer, 2000.

[11] L. Todorovski and S. Džeroski. Experiments in meta-level learning with ilp. In *Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, pages 98–106. Springer, 1999.

Visualizing the Interestingness of Data Mining Results Characterized by Vectors of Probability Distributions

Robert J. Hilderman

Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2
robert.hilderman@uregina.ca

Abstract

Data mining results characterized by vectors of probability distributions are common to many knowledge discovery tasks. When comparing two probability distributions, we frequently use the notion that one of the distributions is somehow more or less diverse (or concentrated) than the other. However, difficulty in determining the diversity of one distribution versus another arises because the measurement of diversity actually consists of two separate components: the number of classes and the proportional distribution of the population among the classes. The Lorenz dominance order has previously been shown to be an effective measure for ranking vectors of proportional distributions in data mining applications. In this paper, we introduce an algorithm that generates rules, based upon the Lorenz dominance order, for constructing a graph that shows the relationship between ranked vectors. Experimental results demonstrate that the graph can provide a reasonable starting point for further subjective evaluation of data mining results.

1. Introduction

The development of measures of interestingness is an active area in KDD. Such measures assist in the interpretation and evaluation of discovered knowledge, and are broadly classified as either subjective or objective. *Subjective measures* are based upon user beliefs or biases regarding relationships in the data, such as an approach utilizing Bayes Rule to revise prior beliefs [10], or an approach utilizing templates to describe interesting patterns [8]. *Objective measures* are based upon the structure of discovered patterns, such as the frequency with which combinations of items appear in sales transactions [1], or results characterized by vectors of probability distributions [5].

Data mining results characterized by vectors of probability distributions are common to many knowledge discovery tasks, such as those that generate generalized relations, data cubes, association rules, and others [5]. A problem that needs to be addressed, then, is how to determine the relative interestingness of vectors. In previous work, we demonstrated a principled approach that uses the Lorenz dominance order to rank vectors in data mining applications [4]. In this work, we continue our study into applications of diversity measures, in particular, we show how the Lorenz dominance order can be used as an aid in visualizing the interestingness of vectors. The relationship that we seek to demonstrate is whether the vectors are comparable. If vector X is more diverse than Y according to the Lorenz dominance order (the criteria to be discussed later), then the vectors are *comparable* and we can say that X *majorizes* Y. Since we consider majorization and interestingness to be equivalent, we say that X is *more interesting* than Y. If X and Y are not comparable according to the Lorenz dominance order, then the relationship between the two vectors remains undefined and we cannot make any determination about their relative interestingness. The reader should note that the Lorenz dominance order is not the same as the order obtained by a topological sort. In a topological sort, the assumption is that all objects being sorted are comparable, whereas comparability is derived mathematically in determining the Lorenz dominance order.

In this paper, we introduce an algorithm that generates rules, based upon the Lorenz dominance or-

der, for constructing a graph that shows the relationship between ranked vectors. Although the general technique is applicable to the results generated by many knowledge discovery tasks, we describe the algorithm within the context of summaries generated by the Multi-Attribute Generalization algorithm [6, 7]. The problem is described, as follows. Let a *summary* S be a relation defined on the columns $\{(A_1, D_1), (A_2, D_2), \ldots, (A_n, D_n)\}$, where each (A_i, D_i) is an attribute-domain pair. Also, let $\{(A_1, v_{i1}), (A_2, v_{i2}), \ldots, (A_n, v_{in})\}$, $i = 1, 2, \ldots, m$, be a set of m unique tuples, where each (A_j, v_{ij}) is an attribute-value pair and each v_{ij} is a value from the domain D_j associated with attribute A_j. One attribute A_k is a derived attribute, called *Count*, whose domain D_k is the set of positive integers, and whose value v_{ik} for each attribute-value pair (A_k, v_{ik}) is equal to the number of tuples which have been aggregated from the base relation (i.e., the unconditioned data present in the original database). We refer to the values in the *Count* column as a *count vector*, or simply, *vector*.

Table 1. A sample summary

Office	*Quantity*	*Amount*	*Count*
West	8	$200.00	4
East	11	$275.00	3

Table 2. A sales transaction database

Office	*Quantity*	*Amount*
2	2	$50.00
5	3	$75.00
3	1	$25.00
7	4	$100.00
1	3	$75.00
6	4	$100.00
4	2	$50.00

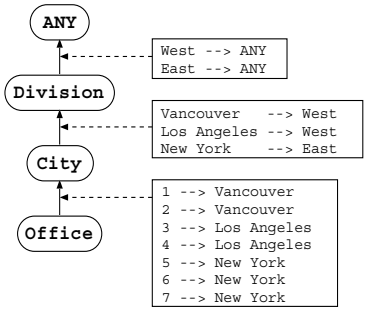


Figure 1. A DGG for the *Office* attribute

A summary, such as the one shown in Table 1, can be generated from a database, such as the one shown in Table 2, using *domain generalization graphs* (DGGs) [6, 7], such as the one shown in Figure 1. For example, the DGG in Figure 1 is associated with the *Office* attribute in the database of Table 2. In Figure 1, the domain for the *Office* attribute is represented by the *Office* node. Increasingly general descriptions of the domain values are represented by the *City*, *Division*, and *ANY* nodes. A user-defined taxonomy in the form of a table is associated with every arc between the nodes in the DGG and describes a generalization relation from one domain to another in a process called *attribute-oriented generalization* (AOG) [3] (other generalization relations besides table lookups are possible, but we restrict our discussion for the sake of simplicity and clarity). The table associated with the arc between the *Office* and *City* nodes defines the mapping of the domain values of the *Office* node to the domain values of the *City* node (e.g., 1 and 2 map to Vancouver, 3 and 4 map to Los Angeles, and 5 to 7 map to New York). The table associated with the arc between the *City* and *Division* nodes can be described similarly. The table associated with the arc between the *Division* and *ANY* nodes maps all values in the *Division* domain to the special value *ANY*. The summary in Table 1 corresponds to the *Division* node of the *Office* DGG, where the corresponding values in the *Quantity* and *Amount* attributes from Table 2 are also aggregated accordingly.

When there are DGGs associated with multiple attributes, then more complex summaries can be generated (known as *multi-attribute generalization*). For example, a DGG for the *Quantity* attribute is shown in Figure 2, where the generalization space consists of three nodes. The set of all possible combinations of domains from the DGGs associated with the *Office* and *Quantity* attributes defines the generalization space for the many summaries that can be generated from Table 2. Thus, the generalization space consists of the 12 nodes shown in Figure 3 (i.e., 4 nodes in the *Office* DGG $\times$ 3 nodes in the *Quantity* DGG), and each node corresponds to a unique summary. For example, the *Division/Quantity* node corresponds to the summary generated by generalizing the *Office* attribute to the level of the *Division* node in the *Office* DGG, while the *Quantity* attribute remains ungeneralized (this summary is equivalent to the summary in Table 1). Similarly, the *City/Status* node corresponds to the summary shown in Table 3, and is generated by generalizing the *Office* and *Quantity* attributes to the level of the *City* and *Status* nodes, respectively.

Naturally, the general technique is applicable to more than two attributes and should now be clear.

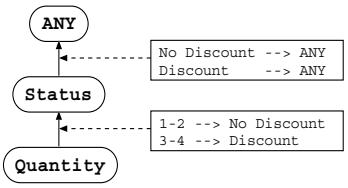


Figure 2. A DGG for the *Quantity* attribute

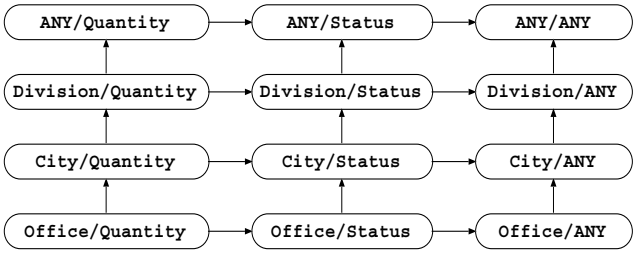


Figure 3. The generalization space defined by the *Office* and *Quantity* DGGs

Table 3. The *City/Status* summary

Office	*Quantity*	*Amount*	*Count*
New York	Discount	$275.00	3
Los Angeles	No Discount	$75.00	2
Vancouver	No Discount	$50.00	1
Vancouver	Discount	$75.00	1

The remainder of this paper is organized as follows. In Section 2, we provide some background on important properties of the Lorenz dominance order as it applies to ranking summaries. In Section 3, we present some experimental results showing how the majorization relationship can be summarized in a graph. We conclude in Section 4 with a brief summary of our work and suggestions for future research.

2. Background

In this section, f denotes a general interestingness measure, such as statistical variance or Shannon's index. In addition, the values in the vectors $(n_1, \ldots, n_m)$ and $(n_1^{'}, \ldots, n_m^{'})$ correspond to the the values in the count vectors associated with two summaries, and are assumed to be arranged in descending order such that $n_1 \geq \ldots \geq n_m$.

The *Lorenz dominance order* [9] compares vectors with different distributions and says for any two vectors $(n_1, \ldots, n_m)$ and $(n_1^{'}, \ldots, n_m^{'})$, that $(n_1^{'}, \ldots, n_m^{'}) \succ (n_1, \ldots, n_m)$ (the $\succ$ character is read as *majorizes*) if the following four conditions hold:

1. $n_1 \geq \ldots \geq n_m$.
2. $n_1^{'} \geq \ldots \geq n_m^{'}$.
3. $\sum_{i=1}^{j} n_i^{'} \geq \sum_{i=1}^{j} n_i$, for every $j = 1, \ldots, m$.
4. $\sum_{i=1}^{m} n_i^{'} = \sum_{i=1}^{m} n_i$.

For example, according to the Lorenz dominance order, $(40, 20, 20, 20) \succ (30, 30, 20, 20)$, but $(40, 20, 20, 20) \not\succ (33, 33, 20, 14)$. Therefore, $(40, 20, 20, 20)$ and $(30, 30, 20, 20)$ are comparable, but $(40, 20, 20, 20)$ and $(33, 33, 20, 14)$ are not.

In earlier work, we described theoretical properties of the Lorenz dominance order as it relates to ranking summaries generated from databases [4]. We repeat these theoretical properties here, without proof, for reader convenience. Please refer to the earlier work for a more complete discussion.

Transfer Principle (P1). Given a vector $(n_1, \ldots, n_m)$, $n_i \geq n_j$, $i < j$, and $0 < c \leq n_j$, then $f(n_1, \ldots, n_i + c, \ldots, n_j - c, \ldots, n_m) > f(n_1, \ldots, n_i, \ldots, n_j, \ldots, n_m)$.

P1, adapted from [2], specifies that when a strictly positive transfer is made from the count of one tuple to another tuple whose count is greater, then interestingness increases. For example, given the vectors $X = (10, 7, 5, 4)$ and $Y = (10, 9, 5, 2)$, where Y is derived from X via a positive transfer of 2 units from the fourth tuple of X to the second tuple of Y, then we require that $f(10, 9, 5, 2) > f(10, 7, 5, 4)$.

Majorization Principle (P2). Given vectors $(n_1, \ldots, n_m)$ and $(n_1^{'}, \ldots, n_m^{'})$, whenever $f(n_1^{'}, \ldots, n_m^{'}) > f(n_1, \ldots, n_m)$, then $(n_1^{'}, \ldots, n_m^{'}) \succ (n_1, \ldots, n_m)$.

Theorem 1. P1 and P2 define a partial order on ranked summaries.

Definition. Let $(n_1^{'}, \ldots, n_m^{'})$ be a vector derived from $(n_1, \ldots, n_m)$ according to P1. That is, for some $n_i \geq n_j$, $i < j$, $0 < c \leq n_j$, we have $(n_1^{'}, \ldots, n_m^{'}) = (n_1, \ldots, n_i + c, \ldots, n_j - c, \ldots, n_m)$. The transfer from n_j to n_i is called *one elementary transfer*.

Theorem 2. Whenever a vector $(n_1^{'}, \ldots, n_m^{'})$ can be derived from a vector $(n_1, \ldots, n_m)$ via a finite

series of elementary transfers, then $(n_1^{'}, \ldots, n_m^{'}) \succ (n_1, \ldots, n_m)$.

Theorem 3. For a summary whose distribution of tuples corresponds to the vector $(n_1, \ldots, n_m)$, if a more general summary resides along the same path in a DGG whose distribution of tuples corresponds to the vector $(n_1^{'}, \ldots, n_m^{'})$, then $(n_1^{'}, \ldots, n_m^{'})$ can be derived from $(n_1, \ldots, n_m)$ via a finite series of elementary transfers.

Theorem 4. P1 and P2 define a total order for the summaries along a single path in a DGG.

Using P1 and P2, we can rank the summaries in the generalization space shown in Figure 3. However, due to space considerations, we will only consider the summaries corresponding to the *Division/Quantity* node (shown in Table 1), the *City/Status* node (shown in Table 3), the *City/Quantity* node (shown in Table 4), and the *Division/Status* node (shown in Table 5). Therefore, our task is to rank the associated count vectors $(4, 3)$, $(3, 2, 1, 1)$, $(3, 2, 2)$, and $(3, 3, 1)$, respectively.

Table 4. The *City/Quantity* summary

Office	*Quantity*	*Amount*	*Count*
New York	11	$275.00	3
Los Angeles	3	$75.00	2
Vancouver	5	$125.00	2

Table 5. The *Division/Status* summary

Office	*Quantity*	*Amount*	*Count*
East	Discount	$275.00	3
West	No Discount	$125.00	3
West	Discount	$75.00	1

In order to rank the count vectors, we need to determine whether the vectors are comparable according to the Lorenz dominance order. Now each vector is sorted in descending order, so in comparing every possible pairing of vectors, conditions 1 and 2 hold. Also, since each summary was generated from the same base relation containing seven tuples, in comparing every possible pairing of vectors, condition 4 holds. To determine whether condition 3 holds, we again need to compare every possible pairing of vectors. It turns out that all the vectors are comparable, so in this simple example, we take advantage of the transitive property of the Lorenz dominance order to avoid the necessity of showing all possible pairings, while still adequately demonstrating the general technique. For example, $(3, 2, 2) \succ (3, 2, 1, 1)$ because $\sum_{i=1}^{j} n_i^{'} \geq \sum_{i=1}^{j} n_i$, for every $j = 1, \ldots, m$. Specifically, $n_1^{'} = 3 \geq n_1 = 3$, $n_1^{'} + n_2^{'} = 5 \geq n_1 + n_2 = 5$, $n_1^{'} + n_2^{'} + n_3^{'} = 7 \geq n_1 + n_2 + n_3 = 6$, and $n_1^{'} + n_2^{'} + n_3^{'} + n_4^{'} = 7 \geq n_1 + n_2 + n_3 + n_4 = 7$. It is okay to refer to $n_4^{'}$ in the last inequality because $(3, 2, 2, 0) \equiv (3, 2, 2)$, and in fact, the *virtual* length of all the vectors is seven (i.e., the number of tuples in the base relation). So, for example, saying $(3, 2, 2) \succ (3, 2, 1, 1)$ is equivalent to saying $(3, 2, 2, 0, 0, 0, 0) \succ (3, 2, 1, 1, 0, 0, 0)$. Continuing with our example, we also have $(3, 3, 1) \succ (3, 2, 2)$ and $(4, 3) \succ (3, 3, 1)$. Applying the transitive property of the Lorenz dominance order, we obtain $(4, 3) \succ (3, 3, 1) \succ (3, 2, 2) \succ (3, 2, 1, 1)$. Consequently, according to this objective measure, ordering the summaries from most to least interesting, we have Table 1 $\succ$ Table 5 $\succ$ Table 4 $\succ$ Table 3.

3. Experimental Results

Input data for the experiments was supplied by the NSERC Research Awards database, freely available in the public domain, and the Customer Accounts database, a confidential database provided by a commercial research partner in the telecommunications industry. The NSERC Research Awards database consists of approximately 10,000 tuples in six tables describing a total of 22 attributes. The Customer Accounts database consists of over 8,000,000 tuples in 22 tables describing a total of 56 attributes. The largest table contains over 3,300,000 tuples representing the account activity for over 500,000 customer accounts and over 2,200 products and services.

A series of experiments were run using *DGG-Majorize*, an extension to *DB-Discover*, a research data mining tool developed at the University of Regina. *DGG-Majorize* evaluates the summaries generated by *DB-Discover* in a two-step process. In the first step, it determines those summaries in which attributes are significantly associated according to the chi-square test for independence, and prunes those in which no significant association is found. In the second step, the remaining summaries are ranked according to the Lorenz dominance order. In the discovery tasks, from two to four attributes were selected for discovery. We refer to the NSERC discovery tasks containing two, three, and four attributes as as *N-2*, *N-3*, and *N-4*, respectively, and the Customer Accounts discovery tasks as *C-2*, *C-3*, *C-4*, respectively.

A summary of the results for the six discovery tasks is shown in Table 6. In Table 6, the *Task* column describes the unique discovery task identifier, the *At-*

Table 6. Summary results for six representative discovery tasks

Task	*Attributes*	*Generated*	*Pruned*	*%Pruned*	*Associated*	*%Associated*
N-2	2	22	14	63.6	8	36.3
N-3	3	70	43	61.4	27	38.6
N-4	4	186	143	76.9	43	23.1
C-2	2	340	325	95.6	15	4.4
C-3	3	3468	3288	94.8	180	5.2
C-4	4	27744	26163	94.3	1581	5.7

Table 7. Summaries and their Lorenz dominance order

ID	*7*	*8*	*11*	*12*	*16*	*17/18*	*21/22*	*27*	*28*	*29/30*	*33/34*	*52*	*53/54*	*57/58*	*79*	*80*	*83*	*84*	*99*	*100*	*123*
7		•	•	•			•		•	•	•			•	•	•	•	•	•	•	
8			•	•							•					•	•	•			
11				•							•						•	•			
12																		•			
16		•	•	•			•		•	•	•	•		•	•	•	•	•	•	•	
17/18	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•	•	•	
21/22			•	•							•			•			•	•			
27		•	•	•					•	•	•			•		•	•	•	•	•	
28				•							•							•		•	
29/30				•					•		•						•	•		•	
33/34																					
52		•	•	•					•	•	•			•		•	•	•	•	•	
53/54		•	•	•			•		•	•	•	•		•	•	•	•	•	•	•	
57/58				•							•						•	•			
79		•	•	•							•					•	•	•			
80				•							•						•	•			
83				•														•			
84																					
99				•					•		•						•	•		•	
100											•							•			
123		•	•	•			•		•	•	•	•		•	•	•	•	•	•	•	

tribute column describes the number of attributes selected, the *Generated* column describes the number of summaries generated, the *Pruned* (*%Pruned*) column describes the number (percentage) of summaries in which no significant association between attributes was found, and the *Associated* (*%Associated*) column describes the number (percentage) of summaries in which a significant association was found and which are available for ranking by the Lorenz dominance order. For example, in *N-3*, an NSERC discovery task, three attributes were selected, 70 summaries were generated, 43 (61.4%) were pruned, and a significant association was discovered between the attributes in the remaining 27 (38.6%) summaries.

The 27 remaining summaries were ranked according to the Lorenz dominance order. These summaries and their Lorenz dominance order are shown in Table 7. In Table 7, the *ID* column describes the unique identifiers associated with each of the 27 summaries, the numbered columns describe those summaries that are majorized by the corresponding summary in the *ID* column, and a summary that is majorized is indicated by the *bullet* symbol (i.e., •). For example, in the second row, it is shown that summary 8 majorizes 11, 12, 33, 34, 80, 83, and 84 (equivalently $8 \succ \{11, 12, 33, 34, 80, 83, 84\}$). Since we consider majorization to be equivalent to interestingness, then essentially we consider summary 8 to be more interesting than 11, 12, 33, 34, 80, 83, and 84. Summaries 33, 34, and 84 are examples of summaries that do not majorize any other summaries.

Summaries whose count vectors were identical (i.e., identical number of tuples and identical probability distributions) are grouped together and treated as a single summary for this analysis (because if vector $X = Y$, then $X \succ Y$ and $Y \succ X$, so the vectors are indistinguishable according to the Lorenz dominance order). For example, the count vectors for summaries 17 and 18 were identical and indistinguishable to the Lorenz dominance order. Consequently, we consider these summaries to be identical, treat them as a single summary for this analysis, and simply refer to the resulting single count vector as 17/18.

Taking advantage of the transitive property of the Lorenz dominance order, we can discover all of the majorization relationships described in Table 7. For example, consider summary 7 in the first row. We see that $7 \succ 8$. Moving to the row beginning with summary 8, we see that $8 \succ 11$. Moving to the row beginning with summary 11, we see that $11 \succ 12$. Moving to the row beginning with summary 12, we see that $12 \succ 84$. Moving to the row beginning with summary 84, we see that

```
1.  procedure Generate_Graph_Description (majorize[ ][ ], summaryCount)
2.  begin
3.      pathCount ← 0
4.      for i = 1 to summaryCount do begin
5.          path[0] ← i
6.          pathCount ← pathCount + Discover_Paths (majorize, path, i, 0, paths, summaryCount)
7.      end for
8.      for i = 1 to pathCount do begin
9.          Delete_Embedded_Paths (i, paths, pathCount)
10.     end for
11.     for i = 1 to pathCount do begin
12.         Report_Common_Path_Segments (i, paths, pathCount)
13.     end for
14. end
```

Figure 4. Generate_Graph_Description procedure

```
1.  procedure Discover_Paths (majorize[ ][ ], path[ ], i, k, paths[ ][ ], summaryCount)
2.  begin
3.      pathCount ← 0
4.      for j = 1 to summaryCount do begin
5.          if majorize[i][j] is true then
6.              k ← k + 1
7.              path[k] ← j
8.              pathCount ← Discover_Paths (majorize, path, j, k, paths, summaryCount) + 1
9.              paths[pathCount] ← path
10.             path[k] ← 0
11.             k ← k − 1
12.         end if
13.     end for
14.     return pathCount
15. end
```

Figure 5. Discover_Paths procedure

84 does not majorize any other summary. Thus, we can summarize the discovered relationship as the partial order $7 \succ 8 \succ 11 \succ 12 \succ 84$. Note that although we know from the first row that $7 \succ \{8, 11, 12, 84\}$, the first row does not tell us anything about the relationships between 8, 11, 12, and 84. We had to examine the rows corresponding to 8, 11, 12, and 84 to discover these relationships.

Using the Generate_Graph_Description procedure shown in Figure 4, we were able to discover all 33332 possible partial orders described by Table 7, and to summarize these majorization relationships in the graph of Figure 6. The Discover_Paths procedure is a recursive procedure that follows the technique described in the previous paragraph for discovering a complete partial order. The Delete_Embedded_Paths and Report_Common_Path_Segments procedures (not shown due to space limitations, but described below) generated 96 rules for consolidating the original 33332 partial orders into the concise graph of Figure 6.

The Delete_Embedded_Paths procedure determines whether a partial order is embedded, either contiguously or non-contiguously, within some other partial order. For example, the partial order $16 \succ 21/22 \succ 12 \succ 84$ is embedded within $16 \succ 21/22 \succ 11 \succ 83 \succ 12 \succ 84$. Any partial order that is found to be embedded within a partial order of greater length is deleted. The Report_Common_Path_Segments procedure determines those partial orders that share common, contiguous majorization relationships. For example, the partial orders $123 \succ 52 \succ 29/30 \succ 28 \succ 100 \succ 84$ and $17/18 \succ 16 \succ 52 \succ 29/30 \succ 28 \succ 100 \succ 84$ share the common path segment $52 \succ 29/30 \succ 28 \succ 100 \succ 84$, where the common segment begins with summary 52. This relationship can be summarized in the graph of Figure 6 by having two paths, $123 \succ 52$ and $17/18 \succ 16 \succ 52$, converge at node 52 and share the common majorization relationships following 52 (i.e., two paths converge into one at node 52).

Using the concise graph of Figure 6, the majorization relationship of the 27 summaries can be easily determined. The shaded nodes with a bold border indicate summaries that are not majorized by any others, and are start points for traversing the graph. For exam-

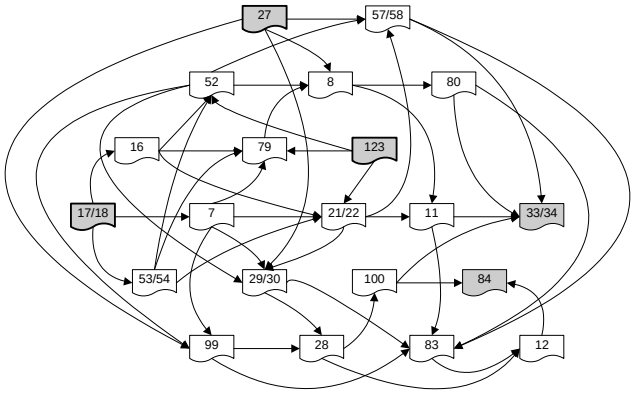


Figure 6. A graph summarizing the Lorenz dominance order

ple, starting at node 17/18, we can follow a path that includes nodes 7, 21/22, 11, and 33/34. Node 33/34 is a shaded node without a bold border, and indicates a stop point (i.e., 33/34 majorizes no other summaries). Similarly, starting at node 17/18, we can follow a path that includes 16, 79, 8, 80, and 33/34. Note that while summary 17/18 majorizes both summaries 7 and 16, there is no path between 16 and 7, so we cannot say anything definitive about the relative interestingness of these two summaries. However, we do know that 17/18 is more interesting than both 16 and 7.

4. Conclusion and Future Research

The Lorenz dominance order compares vectors with different distributions. Here we have shown that it can provide the basis for visualizing the interestingness of summaries generated from databases. With the aid of the Generate_Graph_Description procedure, we were able to summarize the ranking of summaries into a concise graph that described the majorization relationship between all pairs of summaries. The graph generated is an objective evaluation of the relative interestingness of the summaries and provides the domain expert with a starting point for further subjective evaluation.

Future research will focus on the development of effective techniques for automating the construction of a graph from the generated rules. We will also focus on the development of an interactive platform that will enable a domain expert to traverse the graph and display the summaries associated with a node in the graph.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Databases (VLDB'94)*, pages 487–499, Santiago, Chile, September 1994.

[2] H. Dalton. The measurement of the inequality of incomes. *Economic Journal*, 30:348–361, 1920.

[3] J. Han, Y. Cai, and N. Cercone. Data-driven discovery of quantitative rules in relational databases. *IEEE Transactions on Knowledge and Data Engineering*, 5(1):29–40, February 1993.

[4] R.J. Hilderman. The Lorenz dominance order as a measure of interestingness in KDD. In M.-S. Chen, P.S. Yu, and B. Liu, editors, *Proceedings of the 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'02)*, pages 177–185, Hong Kong, May 2002.

[5] R.J. Hilderman and H.J. Hamilton. Applying objective interestingness measures in data mining systems. In D.A. Zighed and J. Komorowski, editors, *Proceedings of the 4th European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'00)*, pages 432–439, Lyon, France, September 2000.

[6] R.J. Hilderman and H.J. Hamilton. *Knowledge Discovery and Measures of Interest*. Kluwer Academic Publishers, 2001.

[7] R.J. Hilderman, Liangchun Li, and H.J. Hamilton. Visualizing data mining results with domain generalization graphs. In U. Fayyad, G.G. Grinstein, and A. Wierse, editors, *Information Visualization in Data Mining and Knowledge Discovery*, pages 251–270. Morgan Kaufmann Publishers, 2002.

[8] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. In N.R. Adam, B.K. Bhargava, and Y. Yesha, editors, *Proceedings of the Third International Conference on Information and Knowledge Management*, pages 401–407, Gaitersburg, Maryland, 1994.

[9] A.W. Marshall and I. Olkin. *Inequalities: Theory of Majorization and its Applications*. Academic Press, 1979.

[10] B. Padmanabhan and A. Tuzhilin. A belief-driven method for discovering unexpected patterns. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*, pages 94–100, New York, New York, August 1998.

General Framework for Mining Frequent Patterns in Structures

Akihiro Inokuchi† INOKUCHI@JP.IBM.COM
Takashi Washio‡ WASHIO@AR.SANKEN.OSAKA-U.AC.JP
Yoshio Nishimura‡ NISHIMURA@AR.SANKEN.OSAKA-U.AC.JP
Hiroshi Motoda‡ MOTODA@AR.SANKEN.OSAKA-U.AC.JP

† Tokyo Research Laboratory, IBM Japan, 1623-14, Shimotsuruma, Yamato, Kanagawa, 242-8502, Japan

‡ Institute of Scientific and Industrial Research, Osaka Univ., 8-1, Mihogaoka, Ibaraki, Osaka, 567-0047, Japan

Abstract

Graph structured data mining to derive frequent subgraphs from a general graph dataset is difficult because the search for subgraphs is combinatorially explosive, and includes subgraph isomorphism matching. Although some approaches have been proposed to derive characteristic patterns from graph structured database, they are to limit the graphs to be searched within a specific class. In this paper, we introduce an approach which achieves the complete search of various classes of frequent subgraphs in a massive labeled graph dataset within highly practical time. The power of our approach comes from the algebraic representation of graphs, its associated operations and well-organized constraints to limit the search space efficiently. Its performance has been evaluated through some real world datasets and the high scalability of our approach has been confirmed for the amount of data and the computation time.

1. Introduction

Graph structured data mining which discovers characteristic subgraph patterns embedded in a general graph dataset is an important problem having broad applications. This problem is very difficult to solve in practical time because the search of possible subgraph patterns is combinatorially explosive, and includes subgraph isomorphism matching which is known to be NP-complete. For example, WARMR is a system to mine frequent subgraphs based on inductive logic programming (ILP) [Dehaspe 98]. It could derive all subgraphs consisting of only a few vertices within tractable time under its direct application to the graph dataset without data preprocessing.

To alleviate the difficulty of the computation time, some other approaches have introduced approximations. The most well known approximation is to apply the greedy search as SUBDUE [Cook 94] and GBI [Yoshida 95, Motoda 97]. However, this is not very suitable to many applications requiring the complete search of the results. Another approach is to limit the graphs to be searched within simpler classes. The levelwise version space algorithm proposed by De Raedt limits the search space to frequent paths for the derivation of the result in tractable time [De Raedt 01, Kramer 01a]. However, this is also weak for some practical use since the class of the structure to be mined is paths but not subgraphs. Although Kramer applied MolFea to HIV dataset to discover path patterns from it, and mentioned relation between discovered patterns and an anti-HIV medicine, it is not easy to imagine chemical compounds from the path patterns described in [Kramer 01b].

To overcome these limitations, we proposed an approach named AGM (Apriori-based Graph Mining) in which the knowledge representation and the search operations are highly dedicated to the graph structure mining [Inokuchi 00, Inokuchi 01]. It can efficiently discover all frequent patterns in terms of induced subgraphs contained in a dataset of labeled graphs. The induced subgraph of a graph G has a subset of the vertices of G and the same edges between pair of vertices as in G. An induced subgraph can be an unconnected graph consisting of some isolated graph fragments. Though AGM is designed to work efficiently in comparison with the aforementioned work, it still requires intractable computation time for many practical scale problems. However, since the interesting patterns in many applications are connected graphs, the limitation of the search to connected subgraphs does not reduce the applicability of the graph structured data mining significantly. FSG which derives the complete set of frequent connected subgraphs included in a given graph dataset has been proposed [Kuramochi 01]. Because this uses a data structure consisting of many address pointers to represent the graph structure and many data ID pointers on working memory in a com-

puter, the scalability on graph size, number of data and data processing speed is quite limited.

In this paper, we propose a general framework for mining frequent patterns in structures. By applying additional specific syntactic biases, it can be easily expanded to a system to derives various types of structures *e.g.* connected graph and/or path structure. We evaluate its performance in terms of the required computation time for the carcinogenic dataset and HIV dataset.

The rest of paper is organized as follows. Section 2 defines graph and frequent graph patterns mining problem. Section 3 describes our general framework for mining frequent patterns in dataset consisting of structures. Section 4 defines some additional specific syntactic biases to derive various types of patterns *e.g.* general graph, connected graph and path patterns. Section 5 provides a experimental evaluation of our algorithm on two real dataset consisting chemical compounds. We provide discussion and related work in section 6 and finally conclude in section 7.

2. Graph and Problem Definitions

The input to frequent graph structured data mining is a set of graphs in which each vertex and each edge have a vertex label and an edge label respectively. When a set $V(G)$ of vertices, a set $E(G)$ of edges, a set $L_V(V(G))$ of vertex labels and a set $L_E(E(G))$ of edge labels are provided as

$$V(G) = \{v_1, v_2, \ldots, v_k\},$$
$$E(G) = \{e_h = (v_i, v_j) | v_i, v_j \in V(G)\},$$
$$L_V(V(G)) = \{lb(v_i) | \forall v_i \in V(G)\},$$
$$L_E(E(G)) = \{lb(e_h) | \forall e_h \in E(G)\}$$

respectively, graph G is expressed as

$$G = (V(G), E(G), L_V(V(G)), L_E(E(G))).$$

where multiple vertices can have an identical label and multiple edges also can have. The number of vertices, $|V(G)|$, is called the size of graph G.

Graph structure can be expressed using an adjacency matrix. Given a graph $G = (V, E, L_V, L_E)$, (i,j)-element $x_{i,j}$ of an adjacency matrix X_k of graph G whose size is k is represented as follows.

$$x_{i,j} = \begin{cases} num(lb(e_h)) & \text{if } e_h = (v_i, v_j) \in E(G) \\ 0 & \text{if } (v_i, v_j) \notin E(G) \end{cases},$$

where $i, j \in \{1, \cdots, k\}$, $num(lb(e_h))$ and $num(lb(v_i))$ are natural numbers assigned to an edge label $lb(e_h)$ and a vertex label $lb(v_i)$ respectively for calculation efficiency. The vertex corresponding to the i-th row (i-th column) of an adjacency matrix is called the i-th vertex, and the graph structure of an adjacency matrix X_k is represented as $G(X_k)$.

The adjacency matrix differs depending on the assignment of rows and columns to vertices of the graph. In other words, an identical graph structure can be represented by multiple adjacency matrices. *Canonical form* is a representative matrix among adjacency matrices which represent an identical graph.

In order to reduce the candidates of the frequent subgraphs, the code of an adjacency matrix is defined as follows. In case of an undirected graph, the code of an adjacency matrix X_k is defined as

$$code(X_k) = x_{1,2}x_{1,3}x_{2,3}x_{1,4}\cdots x_{k-2,k}x_{k-1,k}$$

by using (i,j)-element $x_{i,j}$. In case of a directed graph, it is defined as

$$code(X_k) = c_1 c_2 c_3 \cdots c_{\frac{k(k-1)}{2}}$$

$$c_{\frac{j(j-1)}{2}-(j-i-1)} = (|L_E(E(G))| + 1)x_{j,i} + x_{i,j} \quad (i < j)$$

Furthermore, CODE including the vertex labels is defined as

$$CODE(X_k) = num(v_1)\cdots num(v_k) code(X_k)$$

where it is a concatenation of $num(v_i)$s and $code(X_k)$.

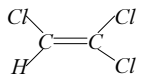


Figure 1. Trichloroethylene

For example a chemical compound is represented by using a graph where each vertex and edge correspond an atom and a chemical bond respectively. When graph shown in Figure 1 is given, where 1, 2 and 3 are assigned to vertex labels H (hydrogen), C (carbon) and Cl (chlorine) respectively, and 0, 1 and 2 are assigned to no bond, a single bond and a double bond, the graph is expressed as

$$X_6 = \begin{array}{c} \\ Cl \\ C \\ Cl \\ C \\ Cl \\ H \end{array} \begin{array}{c} \begin{array}{cccccc} Cl & C & Cl & C & Cl & H \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}.$$

CODE of X_6 is represented as

$$CODE(X_6) = 323231\,101020000100010.$$

Given a graph $G = (V(G), E(G), L_V(V(G)), L_E(E(G)))$, subgraph $G_s = (V(G_s), E(G_s), L_V(V(G_s)), L_E(E(G_s)))$ of G fulfills the following conditions.

$$V(G_s) \subseteq V(G), E(G_s) \subseteq E(G).$$

Given a set GD of graph structured data, the support $sup(G_s)$ of an subgraph G_s is defined as a ratio of the number of graphs including G_s to the total number of graph data in the dataset GD. The derived subgraph having the support more than the *minimum support* specified by a user is called a *frequent subgraph*. A frequent subgraph whose size is k is called a k-frequent subgraph. When the dataset which consists of graph structured data and the minimum support are given, frequent graph structured data mining is to derive all frequent subgraphs that have the support more than or equal to the minimum support value in the dataset[Inokuchi 02].

3. General Framework for Mining Frequent Patterns in Structures

In our previous work, we proposed an approach named AGM (Apriori-based Graph Mining) algorithm in which the knowledge representation and the search operations are highly dedicated to the graph structured data mining [Inokuchi 00]. AGM algorithm can discover not only connected frequent graphs not also unconnected frequent graphs. We use the basic concept of AGM algorithm as a framework of graph structured data mining. By adding some specific syntactic biases to it, AGM framework can discover various types of graphs, *e.g.* connected graphs, path structures and tree structures. In this section, we explain AGM framework, and in next section we introduce additional biases for connected graph derivation and path structure derivation.

AGM framework derives all frequent subgraphs in ascending order of the size of the graph based on the monotonic property that the support of a graph G is less or equal to the supports of its induced subgraph G_s.

$$sup(G) \leq sup(G_s)$$

By using this property, frequent subgraphs are derived stepwisely in ascending order of their sizes beginning with 1-frequent subgraphs. Figure 2 is the outline of our AGM framework. First, a 1×1 adjacency matrix representing a vertex is generated for every vertex, and they are substituted for C_1 (Figure 2 line 1 and 2). Next, the support of the each candidate frequent subgraph is calculated by accessing the database (Figure 2 line 5). Subsequently, Generate-Candidate function generates the candidate frequent subgraphs of size $k+1$ from k-frequent subgraphs in F_k, and they are substituted for C_{k+1} (Figure 2 line 7). Above processes are repeated until C_k becomes empty. Finally, all frequent subgraphs are returned (Figure 2 line 10).

```
// GD is a database consisting of graph structured data.
// F_k is a set of adjacency matrix of k-frequent graphs
// C_k is a set of adjacency matrix of k-candidate graphs
// minsup is minimum support.
 0)  Main(GD, minsup){
 1)    C_1 ←{all adjacency matrices consisting
 2)          of one element};
 3)    k ← 1;
 4)    while(C_k ≠ ∅) {
 5)      Count(GD, C_k);
 6)      F_k ← {c_k ∈ C_k | sup(G(c_k)) ≥ minsup};
 7)      C_{k+1} ← Generate-Candidate(F_k);
 8)      k ← k + 1;
 9)    }
10)    return ⋃_k {f_k ∈ F_k | f_k is canonical}
11)  }
```

Figure 2. Outline of Algorithm

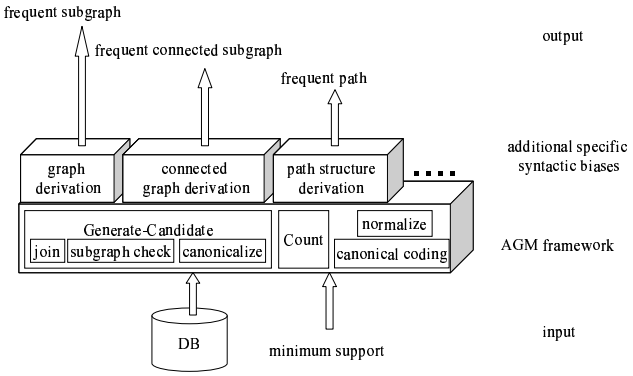


Figure 3. AGM Framework

3.1 Candidate Generation

Generate-Candidate function consists of three parts, join operation, subgraph-check operation and canonicalize operation. In join operation, adjacency matrices of the candidate frequent subgraphs of size $k+1$ are generated by joining two adjacency matrices of k-frequent subgraphs among F_k. Given two adjacency matrices X_k and Y_k representing frequent subgraphs, they are joinable if and only if all of the conditions to join are fulfilled.

<u>*Condition*1</u> X_k and Y_k are identical except the k-th

row and the k-th column, *i.e.*,

$$X_k = \begin{pmatrix} X_{k-1} & \boldsymbol{x}_1 \\ \boldsymbol{x}_2^T & 0 \end{pmatrix}, Y_k = \begin{pmatrix} X_{k-1} & \boldsymbol{y}_1 \\ \boldsymbol{y}_2^T & 0 \end{pmatrix}, \text{and}$$

$lb(v_i \in V(G(X_k))) = lb(v_i \in V(G(Y_k)))\ (i = 1, \cdots, k-1)$.

Condition2 X_k is the canonical form of $G(X_k)$.

If X_k and Y_k are joinable, their *join* operation is defined as follows.

$$Z_{k+1} = \begin{pmatrix} X_{k-1} & \boldsymbol{x}_1 & \boldsymbol{y}_1 \\ \boldsymbol{x}_2^T & 0 & z_{k,k+1} \\ \boldsymbol{y}_2^T & z_{k+1,k} & 0 \end{pmatrix},$$

$lb(v_i \in V(G(Z_{k+1}))) = lb(v_i \in V(G(X_k)))\ (i = 1, \cdots, k-1)$,

$lb(v_k \in V(G(Z_{k+1}))) = lb(v_k \in V(G(X_k)))$, and

$lb(v_{k+1} \in V(G(Z_{k+1}))) = lb(v_k \in V(G(Y_k)))$.

X_k and Y_k are called the first generator matrix and the second generator matrix of Z_{k+1} respectively. Two elements $z_{k,k+1}$ and $z_{k+1,k}$ of Z_{k+1} are not determined by X_k and Y_k. In the case of the undirected graph, the possible graph structures for $G(Z_{k+1})$ are those wherein there is a labeled edge or wherein there is no edge between k-th vertex and $k+1$-th vertex. Then, $(|L_E|+1)$ adjacency matrices under $z_{k,k+1} = z_{k+1,k}$ are generated, where $|L_E|$ is the number of edge labels. The adjacency matrix generated under the above conditions is called a *normal form*.

For the necessary condition of $G(Z_{k+1})$ being a frequent subgraph, all induced subgraphs of $G(Z_{k+1})$ must be frequent subgraphs. The application of this condition reduces the candidates. This is done though subgraph-check operation which is described in detail in the literature [Inokuchi 02].

After generating the matrices of candidate subgraphs, a database is accessed to calculate their supports. However, since multiple normal form matrices can represent an identical graph, the canonical form of these matrices must be identified to collect all counts of the graph. This is done though canonicalize operation which is described in detail in the literature [Inokuchi 00].

3.2 Counting Frequency

After all canonical forms of candidate subgraphs are obtained, the database is accessed, and the frequency of each candidate subgraph is calculated. The simplest subisomorphism matching between a graph of size k and another graph of size K can be achieved by mapping of the vertices from the subgraph to the graph directly. This brute force search can be conducted by the K^k search tree. The worst time complexity of the brute force method turns into exponential time.

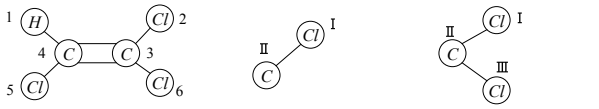


Figure 4. Graph Data and Candidate Subgraphs

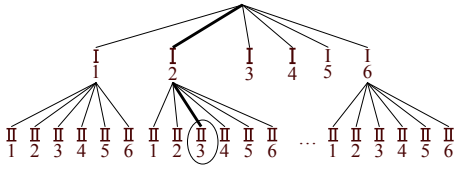


Figure 5. Search Tree for G_K and $G(X_{k-1})$

Let the canonical form of a k-candidate subgraph be X_k, its first generator matrix be X_{k-1}, and a graph in a database whose size is K be G_K. For example, let G_K, $G(X_{k-1})$ and $G(X_k)$ be the graphs of Figure 4(a), (b) and (c) respectively. The canonical forms of Figure 4(b) and (c) are expressed as

$$X_{k-1} = \begin{matrix} \\ Cl \\ C \end{matrix} \begin{matrix} \begin{matrix} Cl & C \end{matrix} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{matrix}, X_k = \begin{matrix} \\ Cl \\ C \\ Cl \end{matrix} \begin{matrix} \begin{matrix} Cl & C & Cl \end{matrix} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{matrix},$$

where $k = 3$. The numbers assigned to the vertices in Figure 4 are vertex IDs. If the brute force method checks whether G_K includes the graph $G(X_{k-1})$ by the depth first search in ascending order of vertex IDs when $G(X_{k-1})$ is the candidate subgraph, it turns out that the graph G_K includes the graph $G(X_{k-1})$, and the correspondences of the vertices between G_K and $G(X_{k-1})$ are 2=I and 3=II, where 2=I shows that vertex whose ID is 2 is mapped to I. The search tree of this case is shown in Figure 5. On the other hand, when $G(X_k)$ is the candidate subgraph, it turns out that graph $G(X_k)$ is included, and the correspondences of

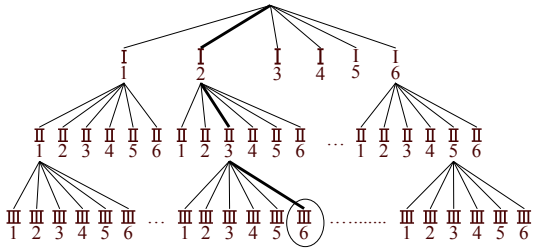


Figure 6. Search Tree for G_K and $G(X_k)$

the vertices are 2=I, 3=II and 6=III as shown in Figure 6. In this case, the search in the part on the left side of the path root-I2-II3 in Figure 6 is not necessary since this part has already been checked in Figure 5. Therefore, if the correspondence relation of the vertex of G_K and $G(X_{k-1})$ is recorded, the G_K's inclusion of the graph structure which has X_{k-1} as the first generator matrix can be efficiently checked. This idea to reuse the matching result in an earlier level enables to calculate frequency much more efficiently than the simple brute force method and some other modified methods of subisomorphism matching [Ullman 76].

4. Additional specific syntactic biases

4.1 Additional bias for General Graph Derivation

The system with this specific syntactic bias corresponds to AGM algorithm which was proposed in [Inokuchi 00].

Canonical Form
We define canonical form as the adjacency matrix which has the minimum (maximum) code among the normal form matrices which represent an identical graph.

Derived Patterns
An algorithm with this specific syntactic bias derives subgraph patterns frequently included in graphs as induced subgraphs. [1] .

Join Operation
The system can derive all frequent subgraphs without any additional conditions for the join operation. However it is possible to reduce search space and derive efficiently all frequent patters by adding the following condition.
Condition3 Given the first generator matrix X_k and the second generator matrix Y_k, when $Code(X_k) \leq Code(Y_k)$ is fulfilled two matrices is joinable[2] .

4.2 Additional bias for Connected Graph Derivation

The interesting patterns in many applications are connected graphs, so the limitation of the search to connected subgraphs does not reduce the applicability of the graph structured data mining significantly. The following is a specific syntactic bias to discover only connected graph patterns in dataset.

Canonical Form
Representing the upper left $i \times i$ submatrix of adjacency matrix X_k as X_i $(1 \leq i \leq k)$, a set $\Gamma(G)$ of adjacency matrices representing an identical graph G will be defined.

$$\Gamma(G) = \{X_k | G(X_i) \text{ is connected, } \forall i = 1, .., k-1, G \equiv G(X_k)\}$$

The adjacency matrix C_k whose CODE is the largest in $\Gamma(G)$ is called the canonical form.

$$C_k \quad w.r.t \quad CODE(C_k) = \max_{X_k \in \Gamma(G)} CODE(X_k)$$

Derived Patterns
The system with this specific syntactic bias derives connected subgraph patterns included as induced subgraphs or subgraph.

Join Operation
Condition3 In case that labels of k-th vertices of $G(X_k)$ and $G(Y_k)$ are identical,

$$code(X_k) \geq code(Y_k).$$

In case that they are not identical,

$$num(lb(v_k \in V(G(X_k)))) > num(lb(v_k \in V(G(Y_k)))),$$

or $G(Y_k)$ is not a connected graph.
Condition4 $G(X_k)$ is a connected graph.

4.3 Additional bias for Path Derivation

Canonical Form
Canonical form is defined similar to connected graph specific syntactic bias.

Derived Patterns
An algorithm with this specific syntactic bias derives subgraph patterns included as paths which have no loops and/or branches.

Join Operation
The following condition is added to condition 1, 2, 3 and 4 for connected graph specific syntactic bias.
Condition5 $G(Z_{k+1})$ has no loops and branching.

5. Experiment

IBM PC 300PL with Windows 2000 was used for the experiments where PentiumIII-667MHz and 192MB of main memory are installed.

5.1 Carcinogenic Data

The molecular structure data of carcinogenic compounds were analyzed. This data was provided by

[1] It is possible to derive the patterns included as a subgraph but as an induced subgraph. However, because the patterns consisting of some vertices with no edges or few edges are derived and it is difficult to understand them, discovered structures are defined as patterns included as an induced subgraph.

[2] In the case that a canonical form is the maximum code, $Code(X_k) \geq Code(Y_k)$.

Predictive Toxicology Evaluation [PTE], and contains information on 340 chemical compounds. The number of types of the atoms which constitute chemical compounds is 24. In addition, the atoms take some different states, and thus the total number of atom types is 66. The atomic bonds which correspond to edges in a graph have 4 types. The average size of the graph data is around 27, and the maximum size is 214.

Figure 7 shows the result of computation time for various minimum support values. It includes the results of AGM for both a connected subgraph derivation and a connected induced subgraph, and FSG. Experiments of FSG were done on dual AMD Athlon MP 1800+ machines with 2GM main memory, running the Linux operating system [Kuramochi 02]. AGM for the both derivations far outperforms FSG.

Figure 8 shows two examples of derived frequent subgraphs in the connected induced subgraph derivation. 6 chemical compounds with carcinogenic activity and 19 compounds without the activity contain the induced subgraph depicted in Figure 8(a). Similarly, 17 compounds with carcinogenic activity and 4 compounds without the activity contain the induced subgraph in Figure 8(b). The former molecular substructure does not induce significant activity whereas the latter induces quite high activity.

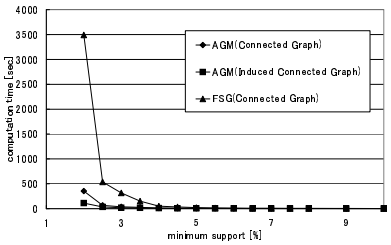


Figure 7. Minimum Support v.s. Computation Time

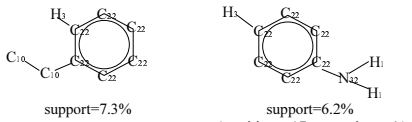


Figure 8. Extracted Frequent Connected Graphs

5.2 HIV Data

The Developmental Therapeutics Program's AIDS Antiviral Screen has checked tens of thousands of compounds for evidence of anti-HIV activity[HIV]. Available are screening results and chemical structural data on compounds that are not covered by a confidentiality agreement. The dataset contains 42,687 chemical compounds' structures and screening data. Each data is categorized into one class of active (CA), moderately active (CM) and inactive (CI). 422 compounds are categorized into CA and 1,081 compounds are categorized to CM. The number of types of the atoms which constitute chemical compounds is 68, hydrogen atoms are omitted and atoms with aromatic bonds are distinguished atoms with no aromatic bonds. The atomic bonds which correspond to edges in a graph have 4 types. The average size of the graph data is around 25.6, and the maximum size is 222. Kramer applied MolFea to 41,768 compounds to discover characteristics path patterns (called fragment) [Kramer 01b]. Our dataset is slightly different than that used in MolFea. This is because that Kramer used the tool *babel* to convert the dataset and it could correctly convert 41,768 chemical compounds. For the first task, fragments which were contained in CA compounds more than 13 which corresponds that the minimum support on the CA dataset is 3% and in CI less than 516 (1.282%) which was chosen based on the χ^2 statistic were mined. The total computation time was about five hours and twenty minutes, and fragments more than 1,600 were discovered. For the second task, fragments which were contained in 13 CA compounds more than and in CM compounds less than 8 (0.8%) were derived. The total computation time was about 34 minutes and fragments more than 680 were discovered.

We applied our graph mining algorithm to all the HIV dataset. After our system finds all frequent subgraphs with supports more than minimum support on dataset consisting of one class, the patterns more than maximum support on the dataset consisting of another class are deleted. First we used AGM framework with path derivation specific syntactic bias. For our first task, we set minimum support on CA dataset and maximum support on CI to 3% (13 compounds) and 1.266% (512 compounds) respectively which are chosen similar to MolFea. The total computation time was about 10 minutes. For our second task, we set minimum support on CA dataset and maximum support on CM to 3% (13 compounds) and 0.8% (8 compounds). It took about one minute. Our algorithm can much more quickly extract all fragments which fulfill the minimum and maximum support constraints than MolFea.

Next we applied our system with connected graph derivation specific syntactic bias to the all dataset. For the first task, a minimum and maximum support is set to 3.6% (16 compounds) and 1.698% (699 compounds) respectively. It took about 3 hours and half and 770,000 patterns were derived. For the second task, a minimum and maximum support is set to 3.6% (16 compounds) and 1.2% (12 compounds) respectively. It took about 3 hours and about 730,000

patterns were derived. Although search space on frequent connected graph derivation is much larger than that on frequent path derivation, AGM can discover all frequent graphs in a practical time. Figure 9 is one of patterns which were discovered in the first task. It is contained in 64 compounds whose classes are CA and 16 compounds whose classes are CI. It is contained by azidothymidine and its similar compounds.

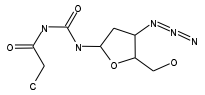


Figure 9. Extracted Frequent Connected Graph

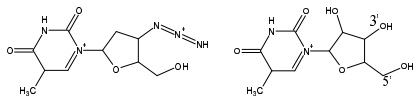


Figure 10. Azidothymidine and Thymine

When viruses invade from the exterior, an immunity system works to eliminates them in human body. CD4 cell plays the central role of the immunity system. HIV invades into CD4 cells, increases and destroys them. After infection by HIV, within CD4 cell, RNA of HIV is replicated into DNA by reverse transcriptase, and is included in a host chromosome. Although medical treatment is difficult because of a capability to hide in the host chromosome of HIV, since the reverse transcriptase of HIV is unnecessary for the host cell, reverse transcriptase is made into the target in development of anti-HIV medicine. The azidothymidine (AZT, Figure 10(a)) which is known to be anti-HIV medicine and whose structure is similar to Thymine (Figure 10(b)) connects to the part with reverse transcriptase and is added to DNA chain under extension. Since it has no hydroxyl groups (-OH) at 3' end of AZT, it stops compounding DNA any more. Our system discovered other characteristic patterns with statistical significance and they are possible to develop new anti-HIV medicine.

6. Discussion and Related Work

Some heuristic based approaches, *e.g.* SUBDUE [Cook 94] and GBI [Yoshida 95, Motoda 97], or some limitation on the class of subgraphs in search [De Raedt 01] have been introduced to alleviate the complexity issue. SUBDUE derives characteristic patterns based on Minimum Description Length of subgraphs. The recent version of GBI derives characteristic patterns in a dataset by chunking pair of connected vertices having a high score [Matsuda 00]. The advantage of these methods is the ability to search typical patterns under various criteria in rapid manner. However, their greedy search may miss some important patterns. De Raedt et al. proposed an approach to derive frequent paths in graphs [De Raedt 01]. Although the computation time to derive the paths by MolFea is the same as that to discover the connected subgraphs, if the class of the structure to be mined is paths but not subgraphs, AGM can discover them much faster than MolFea. MolFea is able to employ syntactic constraints. In [De Raedt 01], generality and specialty among path patterns (fragments) are defined, user can get not only patterns which have the support value greater than or equal to minimum support but also patterns which contain or do not contain a specific structure. Similarly, it is able to define them among graph patterns and to use the similar constraints in our approach.

In contract, WARMR [Dehaspe 98], and FSG [Kuramochi 01] use the complete search similarly to AGM. WARMR derives frequent patterns in a graph dataset by using level-wise search. The patterns are represented by the first order predicates. The basic algorithm structure of FSG is similar to AGM. However, the knowledge representation of graphs and its associated operations are not tuned to graphs. As shown in Figure 7, AGM with a connected graph derivation specific syntactic bias can derive the complete result within a few minutes where this performance is far faster than the other approaches. This is because the representation of graphs and the associated operations are well-organized and dedicated to the mining of graphs. This performance level is considered to be highly practical in many applications.

Recently Zaki[Zaki 02] and Asai et. al.[Asai 02] have proposed methods to find frequent patterns in tree structured dataset. Although they discover efficiently all frequent subtree patters, it is limited to ordered trees in dataset. However additional specific syntactic biases for ordered tree and general tree derivation can be easily implemented on our AGM framework. In the case of the general tree, each tree data is represented directed graph where each vertex (node) have directed edge to parent node, and the condition that generated candidate graph has no loop is added to conditions for connected graph specific syntactic bias. We plan to apply the bias for the tree derivation to investigate ability of it.

7. Conclusion

We proposed a general framework for graph structured data mining. By additional specific syntactic biases,

it can be easily expanded to system to derive various types of structures. We evaluated its performance in terms of the required computation time for the real world datasets. Computational efficiency which is superior to the other approaches has been confirmed.

Acknowledgement

We would like to thank Prof. Luc De Raedt and Dr. Stefan Kramer of University of Freiburg, Prof. Takashi Okada of Kwansei University and Toshiro Takase of IBM Tokyo Research Laboratory for their help and advice.

References

[Asai 02] Asai, T., Abe, K., Kawasoe, S., Arimura, H., Sakamoto, H. & Arikawa, S. Efficient Substructure Discovery from Large Semi-Structured Data. *Proc. of the 2nd SIAM International Conference on Data Mining*, (2002), pp. 158–174.

[Cook 94] Cook, D. J., & Holder, L. B. Substructure Discovery Using Minimum Description Length and Background Knowledge. *Journal of Artificial Intelligence Research*,(1994), Vol.1, pp. 231–255.

[Dehaspe 98] Dehaspe, L., Toivonen, H., & King, R. D. Finding frequent substructures in chemical compounds. *Proc. of the 4th International Conference on Knowledge Discovery and Data Mining*, (1998), pp. 30–36.

[De Raedt 01] De Raedt, L., & Kramer, S. The Levelwise Version Space Algorithm and its Application to Molecular Fragment Finding. *Proc. of the 17th International Joint Conference on Artificial Intelligence*, (2001), pp. 853–859.

[Deshpande 02] Deshpande, M., Kuramochi, M. & Karypis, G. Automated Approaches for Classifying Structures, *Proc. of the 2nd Workshop on Data Mining in Bioinformatics*, (2002).

[HIV] AIDS Antiviral Screen, http://dtp.nci.nih.gov/docs/aids/aids_data.html

[Inokuchi 00] Inokuchi, I., Washio, T., & Motoda, H. An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. *Proc. of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases*, (2000), pp. 13–23.

[Inokuchi 01] Inokuchi, A., Washio, T., Okada, T., & Motoda, H. Applying the Apriori-based Graph Mining Method to Mutagenesis Data Analysis. *Journal of Computer Aided Chemistry*, (2001), Vol.2, pp. 87–92.

[Inokuchi 02] Inokuchi, I., Washio, T., Nishimura, Y., & Motoda, H. A Fast Algorithm for Mining Freqent Connected Graph. *IBM Research Report* RT0448, 2002, February.

[Kramer 01a] Kramer, S., & De Raedt, L. Feature Construction with Version Space for Biochemical Applications. *Proc. of the 18th International Conference on Machine Learning*, (2001), pp. 258–265.

[Kramer 01b] Kramer, S., De Raedt, L., & Helma, C. Molecular Feature Mining in HIV data. *Proc. of the 17th International Conference on Knowledge Discovery and Data Mining*, (2001), pp. 136–143.

[Kuramochi 01] Kuramochi, M., & Karypis, G. Frequent Subgraph Discovery. *Proc. of the 1st IEEE International Conference on Data Mining* (2001), pp. 313–320

[Kuramochi 02] Kuramochi, M., & Karypis, G. An Efficient Algorithm for Discovering Frequent Subgraphs. *Technical Report* 02-026, (2002) .

[Matsuda 00] Matsuda, T., Horiuchi, T., Motoda, H., & Washio, T. Extension of Graph-Based Induction for General Graph Structured Data. *Proc. of the 4th Pacific- Asia Conference on Knowledge Discovery and Data Mining*, (2000), pp. 420–431.

[Motoda 97] Motoda, H., & Yoshida, K. Machine Learning Techniques to Make Computers Easier to Use. *Proc. of the 15th International Joint Conference on Artificial Intelligence*, (1997), Vol. 2, pp. 1622–1631.

[PTE] PTE http://oldwww.comlab.ox.ac.uk/oucl/groups/machlearn/PTE

[Ullman 76] Ullman, J. R. (1976). An algorithm for subgraph isomorphism, *Journal of the ACM*, Vol. 23, no. 1, pp. 31–2.

[Yoshida 95] Yoshida, K., & Motoda, H. CLIP: Concept Learning from Inference Patterns. *Artificial Intelligence*, (1995). Vol. 75, No. 1 pp. 63–92.

[Zaki 02] Zaki, M. Efficintly Mining Frequent Trees in a Forest. *Proc. of the 18th International Conference on Knowledge Discovery and Data Mining*, (2002).

Kernels for Graph Classification

Hisashi Kashima and Akihiro Inokuchi
IBM Research, Tokyo Research Laboratory
1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502, Japan
{hkashima, Inokouchi}@jp.ibm.com

Abstract

In this paper, we apply kernel methods to graph classification problems. To achieve the goal, we have to design an appropriate kernel for computing inner products for pairs of graphs represented in a feature space. We define a graph kernel by a random walk on a vertex product graph of two graphs. Some experiments on predicting properties of chemical compounds show encouraging results.

1 Introduction

Recently, it is needed to develop various kinds of data mining methods that can handle structural data. As semi-structured data such as XML and HTML are increasing, data mining methods that can handle not only relational data, but also for semi-structured data are attracting considerable attention. In pharmaceutical area, it is valuable for rationalization of drug discovery processes to predict the effectiveness or toxicity of drugs from their chemical structures since we can evaluate candidate compounds before we synthesize them.

In this paper, we aim to develop solutions to classification problems of graphs with vertex labels and edge labels (Figure 1). For example, semi-structured data and chemical compounds stated above can be represented as such graphs naturally.

In general learning problems, objects are represented as vectors in a feature space, and training classifiers is reduced to deciding on rules to separate vectors that belong to positive examples from vectors that belong to negative examples. However, when we handle more complex objects such as sequences, trees, and graphs that have structures among their constituent elements, design of a suitable feature space is not trivial. Probably, one of the sound strategies for handling such complex objects is to use local structures in them as features. However, in most cases, considering all possible local structures as features is inhibitive since it often leads to combinatorial explosion. Therefore, some mechanism is needed to select a subset of local structures that can contribute to classification. Relational learning [13] is a general method that can handle local structures in objects. In relational learning, several relationships among constituent elements are defined, and the relationships constitute local structures. The local structures used as features are incrementally built up in the process of training. However, since the problem of finding the best hypothesis is generally NP-hard, we must use heuristic methods. Another method is based on pattern discovery algorithms that find local structures appearing frequently [11, 6], and these structures are used as features. The pattern-discovery-based method has an advantage in that it can make use of unlabelled data. However, the process of discovering patterns is again almost always NP-hard.

Yet another approach is to use kernel methods such as support vector machines (SVMs) [15]. One of the important properties of kernel methods is their access to examples via kernels. In kernel methods, examples are mapped into a feature space implicitly, and only the inner products of the vector representations are used when learning machines access the examples. This means that even in cases where the dimension of the vector representations is extremely high, the dimensions do not explicitly appear in the process of training and classification as long as an efficient procedure to compute the inner products is available. The function giving the inner products is called the 'kernel', and kernel methods can work efficiently in high dimensional feature spaces by using kernels. Moreover, SVMs are known to have good generalization properties, both theoretically and experimentally, and overcome the 'curse of dimensionality' problem in high dimensional feature spaces [15].

Now, our task is to design suitable kernels that can classify structural objects, and that can be computed efficiently. We need a kernel function $K(G_1, G_2)$ that can be efficiently computed the inner product of two vectors represent two graphs G_1 and G_2 in a suitably defined feature space where graphs can be classified. There are several works that aim at classification of structural objects. Haussler [4] introduced 'convolution kernels', a general framework for han-

dling discrete data structures by kernel methods. In the context of the convolution kernels, Watkins [16] and Leslie et al. [12] proposed kernels for strings, and Collins et al. [1] and Kashima et. al. [9] proposed kernels for trees. Besides, Jaakkola et al. [7] proposed Fisher kernels that define kernels using given probabilistic models, and apply them to classification of protein sequences. Of special interest here, Kandola et al. [8] proposed diffusion kernels that define kernels when input spaces are represented as undirected graphs. They employed the idea of diffusion over given graphs to define similarity between arbitrary two vertices. Kondor et al. [10] applied this idea to document classification, where a document corresponds to a vertex. In diffusion kernels, graphs represent the structures of the input spaces, and the vertices are the objects to be classified, while in this paper, our aim is to classify graphs themselves.

To define a kernel between arbitrary two graphs, we use a random walk on the vertex product graph of the two graphs. Precisely, the kernel is defined to be the probability with which two label sequences generated by two 'synchronized' random walks on the graphs are identical. In the feature space, each feature of the vector representation of a graph corresponds to a particular label path that can possibly be generated by a random walk on the graph. Although it is inhibitive to compute inner products explicitly since the number of possible paths is exponentially large, we show simultaneous linear equations to compute them. Therefore, we can compute the kernels efficiently by methods such as iterative methods.

Our kernel is closely related to the diffusion kernels, and we can show their structures are similar. However, while diffusion kernels are defined by a symmetric adjacent matrix which represents the input space graph, our kernel is defined by an asymmetric matrix which represents the vertex product graph of two graphs.

Finally, we perform some experiments on predicting properties of chemical compounds to investigate how our kernel performs well on real data, and the results show encouraging results.

This paper is organized as follows. In Section 2, we define our task, and introduce the idea of kernel methods. In Section 3, we propose a new kernel for graph classification. In Section 4, we summarize the results of our experiments on classification of chemical compounds. We conclude with Section 5 in which we provide a summary and duscussion.

2 Graph Classification Problems and Kernel Methods

In this section, we define the graph classification problem, and introduce the idea of the kernel methods. We define a graph classification problem as the followings. A learning machine receives a set of N training examples

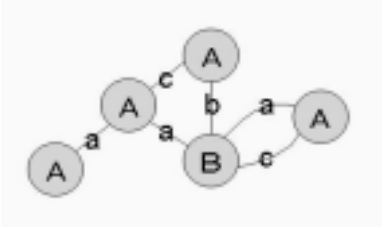


Figure 1. an example of graphs with labels

$\{(G_1, y_1), (G_2, y_2), \ldots, (G_N, y_N), \}$ where each example (G_i, y_i) is given as a pair of a graph $G_i = (V_i, E_i)$ and the class $y_i \in \{+1, -1\}$ that the graph belongs to. We assume that each vertex $v \in V_i$ is labeled by one of the possible vertex labels $\Sigma_V = \{\sigma_{V1}, \sigma_{V2}, \ldots\}$, and each edge is labeled by one of the possible edge labels in Σ_E. Figure 1 shows an example of graphs that we treat in this paper. The objective of the learning machine is to correctly predict the classes of test examples whose classes are unknown.

In this paper, we employ kernel methods for this task. One of the important properties of kernel methods is their access to examples via kernels. In kernel methods, examples are mapped into a feature space implicitly, and only the inner products of the vector representations are used when learning machines access the examples. For example, in the support vector machine that is a well-known kernel learning algorithm, training a classifier is reduced to the following quadratic programming problem,

$$\begin{aligned} \underset{\alpha_1,\ldots,\alpha_N}{\text{maximize}} \quad & \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j < \mathbf{x}_i, \mathbf{x}_j > \\ \text{s.t.} \quad & \alpha \geq 0 \\ & \sum_{i=1}^{N} \alpha_i y_i = 0 \end{aligned} \tag{1}$$

where $\mathbf{x}_i$ is the vector representation of the i-th training example. A text example $\mathbf{x}$ is classified by

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{N} \alpha_i y_i < \mathbf{x}, \mathbf{x}_i > +b\right). \tag{2}$$

We can see all the accesses to the examples are done by inner products. This means that even in cases where the dimension of the vector representations is extremely large, the dimension does not explicitly appear in the process of training and classification as long as an efficient procedure to compute the inner products is available. The function giving the inner products is called 'kernel', and kernel methods can work efficiently in high dimensional feature spaces by using kernels. Therefore, now, our task is to design a suitable feature space where graphs can be classified, and to

give a kernel function $K(G_1, G_2)$ that can efficiently compute the inner product of the two vector representations of two graphs G_1 and G_2.

3 Kernels for Graph Classification

3.1 Graph Kernels

Probably, the most simplest way of defining a vector representation X_G of a graph $G = (V, E)$ is to define each element of a vector using the number of times a particular vertex label appears in the graph.

$$\mathbf{x}_G = \left(\frac{\#(\sigma_{V1}, G)}{|V|}, \frac{\#(\sigma_{V2}, G)}{|V|}, \ldots, \frac{\#(\sigma_{V|\Sigma_V|}, G)}{|V|} \right) \quad (3)$$

where $\#(\sigma_{Vi}, G)$ is the number of times vertex label σ_{Vi} appears in graph G. This corresponds to the bag-of-words representation of a document which is usually used in information retrieval [2]. Suppose we want to calculate the kernel for a pair of graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Then the kernel is defined as

$$K(G_1, G_2) := \mathbf{x}_{G_1} \mathbf{x}_{G_2}^T \quad (4)$$

$$= \frac{1}{|V_1||V_2|} \sum_{v_1 \in V_1} \sum_{v_2 \in V_2} k(v_1, v_2) \quad (5)$$

$$k(v_1, v_2) := I(v_1, v_2) \quad (6)$$

where I is an indicator function that returns 1 when the labels of two arguments are identical, and returns 0 otherwise. The kernel for graphs can be considered to be decomposed in the kernels of pairs of vertices. The vertex-wise kernel of $k(v_1, v_2)$ checks if the labels of the vertices v_1 and v_2 are identical, and this can be seen as a kind of similarity between two vertices. However, $k(v_1, v_2)$ does not incorporate any local information around v_1 and v_2 at all, and therefore we modify $k(v_1, v_2)$ to consider the local structure of graphs. We redefine $k(v_1, v_2)$ so as to take higher scores when not only the labels of v_1 and v_2 are identical, but also the labels of the edges and vertices adjacent to v_1 and v_2, and the further edges and vertices are identical. Concretely, we redefine $k(v_1, v_2)$ as

$$k(v_1, v_2) := (1-\lambda) \cdot k_0(v_1, v_2) \quad (7)$$
$$+\lambda(1-\lambda) \sum_{\substack{e_1 \in A(v_1) \\ e_2 \in A(v_2)}} k_1(v_1, v_2, e_1, e_2)$$
$$+\lambda^2(1-\lambda) \sum_{\substack{e_1 \in A(v_1) \\ e_2 \in A(v_2)}} \sum_{\substack{e_1' \in A(\delta(v_1, e_1)) \\ e_2' \in A(\delta(v_1, e_1))}} k_2(v_1, v_2, e_1, e_2, e_1', e_2')$$
$$+\lambda^3(1-\lambda) \sum_{\substack{e_1 \in A(v_1) \\ e_2 \in A(v_2)}} \sum_{\substack{e_1' \in A(\delta(v_1, e_1)) \\ e_2' \in A(\delta(v_2, e_2))}} \sum_{\substack{e_1'' \in A(\delta(\delta(v_1, e_1), e_1')) \\ e_2'' \in A(\delta(\delta(v_2, e_2), e_2'))}} k_3(v_1, v_2, e_1, e_2, e_1', e_2', e_1'', e_2'')$$
$$+ \cdots$$

$$k_0(v_1, v_2) = I(v_1, v_2) \quad (8)$$

$$k_1(v_1, v_2, e_1, e_2) = k_0(v_1, v_2) \cdot \frac{I(e_1, e_2) \cdot I(\delta(v_1, e_1), \delta(v_2, e_2))}{|A(v_1)||A(v_2)|} \quad (9)$$

$$k_2(v_1, v_2, e_1, e_2, e_1', e_2') = k_1(v_1, v_2, e_1, e_2) \cdot \frac{I(e_1', e_2') \cdot I(\delta(\delta(v_1, e_1), e_1'), \delta(\delta(v_2, e_2), e_2'))}{|A(v_1')||A(v_2')|} \quad (10)$$

$$k_3(v_1, v_2, e_1, e_2, e_1', e_2', e_1'', e_2'') = k_2(v_1, v_2, e_1, e_2, e_1', e_2') \cdot \frac{I(e_1'', e_2'') I(\delta(\delta(\delta(v_1, e_1), e_1'), e_1''), \delta(\delta(\delta(v_2, e_2), e_2'), e_2''))}{|A(v_1'')||A(v_2'')|} \quad (11)$$
$$\cdots$$

where $\lambda \in [0, 1]$ is a decaying constant, and $A(v)$ is a set of edges adjacent to v, and $\delta(v, e)$ is a transition function that returns the vertex at the other side of e adjacent to v. Note that Equation (7) is identical to Equation (6) when $\lambda = 0$. The new kernel $K(G_1, G_2)$ with modified $k(v_1, v_2)$ can be interpreted using a random walk on the vertex product graph $G_{1\times2} = (V_1 \times V_2, E_{1\times2} \subseteq E_1 \times E_2)$ of two graphs G_1 and G_2. Suppose that two 'synchronized' random walks are performed on G_1 and G_2 as the following. At first $v_1 \in V_1$ and $v_2 \in V_2$ are selected randomly as the starting points. At each round, both random walks are simultaneously halted with probability $1 - \lambda$, and continued with probability λ. If continued, in each graphs, a transition is made by randomly selecting an edge among the edges adjacent to the current vertex. When the random walks are halted, the trial succeeds if the two label sequences generated from two graphs are identical. $K(G_1, G_2)$ can be interpreted as the probability with that this trial succeeds.

From the viewpoint of constructing a feature space, each feature of graph G is the probability with which a particular label path is generated by a (single) random walk on G with halting probability λ.[1] Explicit Computation of Equation (7) is inhibitive because of the exponentially many label sequences. However, we can rewrite the equation as the following linear equations.

$$k(v_1, v_2) = I(v_1, v_2) \left\{ (1-\lambda) + \lambda \sum_{\substack{e_1 \in A(v_1) \\ e_2 \in A(v_2)}} \frac{I(e_1, e_2)}{|A(v_1)||A(v_2)|} \cdot k(\delta(v_1, e_1), \delta(v_2, e_2)) \right\} \quad (12)$$

[1] For deriving Equations (7)-(11), each feature should be multiplied by $\sqrt{\frac{1-\lambda}{\lambda^i}}$ where i is the length of the random walk that generate the label sequence. However, these factors do not influence the kernel values after appropriate scaling.

3.2 Relation to Diffusion Kernels

In this subsection, we discuss the relationship between our kernel and von Neumann kernel proposed by Kandora et al. [8]. This kernel is a kind of diffusion kernels introduced by Kondor et al. [10], and Kandora et al. [8] applied the idea of diffusion kernels to document classification. They regard a document as a vertex in a graph, and represent the similarity between two documents as the weight of an edge. In other words, suppose that K is a symmetric adjacent matrix of the graph, the (i,j)-th element indicates the 'direct' similarity defined to be the inner product of the two bag-of-words vector representations of i-th document and the j-th document. Although K itself is a kernel matrix defined over the vertices as long as K is positive definite, they defined von Neumann kernel as

$$\begin{aligned} K_{vN} &:= L + \lambda K K_{vN} && (13)\\ &= K + \lambda K^2 + \lambda^2 K^3 + \cdots && (14)\\ &= (I - \lambda K)^{-1} K && (15) \end{aligned}$$

to incorporate the 'indirect' similarities. Intuitively, this kernel implements the idea that two documents are similar if both of them are similar to another document. The (i,j)-th element of K^d indicates the sum of the products of the edge weights in all possible paths of length d between the i-th vertex and the j-th vertex. Note that the paths can include a particular edge more than once.

Similarly, we can rewrite Equation (12) by matrices. Let $\mathbf{k}$ be a vector whose dimension is $|V_1| \cdot |V_2|$, and whose $i_{v_1 v_2}$-th element is $I(v_1, v_2)$ be $k(v_1, v_2)$ where $i_{v_1 v_2}$ is the index for (v_1, v_2). Similarly, let $\mathbf{k}_0$ be a vector whose $i_{v_1 v_2}$-th element is $I(v_1, v_2)$. Using the $|V_1| \cdot |V_2| \times |V_1| \cdot |V_2|$ matrix K' defined as

$$[K']_{i_{v_1 v_2}, i_{v_1' v_2'}} := \sum_{\substack{e_1 \in A(v_1) \\ \delta(v_1, e_1) = v_1'}} \sum_{\substack{e_2 \in A(v_2) \\ \delta(v_2, e_2) = v_2'}} \frac{I(e_1, e_2)}{|A(v_1)||A(v_2)|}, \quad (16)$$

we rewrite Equation (12) as

$$\begin{aligned} \mathbf{k} &= (1-\lambda)\mathbf{k}^0 + \lambda K' \mathbf{k} && (17)\\ &= (1-\lambda)(\mathbf{k}^0 + \lambda K' \mathbf{k}^0 + \lambda^2 K'^2 \mathbf{k}^0 + \cdots) && (18)\\ &= (1-\lambda)(I - \lambda K')^{-1} \mathbf{k}^0. && (19) \end{aligned}$$

Apparently, Equations (13)-(15) and (17)-(19) have a common structure, and both can be interpreted as a random walks on graphs. However, we point some differences between them. The diffusion kernels are defined over undirected graphs, that is, K is symmetric. On the other hand, our kernel is defined over directed graphs, that is, K' is asymmetric. Moreover, in diffusion kernels, a kernel function defines the similarity between an arbitrary pair of objects in the input space represented as a graph, while in our kernel, an object itself is a graph, and the similarity between two arbitrary pairs of vertices is defined over the vertex product graph $G_{1\times 2} = (V_1 \times V_2, E_{1\times 2} \subseteq E_1 \times E_2)$ of two graph G_1 and G_2. In other words, while the (i,j)-th element of $(I - \lambda K)^{-1} K$ indicates the similarity between the i-th vertex and the j-th vertex, the $(1-\lambda)(i_{v_1 v_2}, i_{v_1' v_2'})$-th element of $(I - \lambda K')^{-1}$ indicates the contribution from the similarity of vertex pair v_1' and v_2' to the similarity of vertex pair v_1 and v_2.

4 Experiments

In this section, we apply our kernel to prediction of the properties of chemical compounds. A chemical compound can be represented as a graph by considering the names of atoms as vertex labels, and the types of bonds as edge labels. We used two datasets, mutag dataset [14] and PTC dataset [5]. In mutag dataset, the task is to predict mutagenicity, and 188 compounds are included, and the maximum number of vertices is 40, and the average number of vertices is 31.4. In PTC dataset, the task is to predict carcinogenicity, and 417 compounds are included, and the maximum number of vertices is 109, and the average number of vertices is 25.7. Each compound in PTC dataset is given four classes, MM(Male Mouse), FM(Female Mouse), MR(Male Rat) and FR(Female Rat), each of which takes one of $\{\text{EE}, \text{IS}, \text{E}, \text{CE}, \text{SE}, \text{P}, \text{NE}, \text{N}\}$. Therefore, PTC provides four classification problems. We use $\{\text{CE}, \text{SE}, \text{P}\}$ as positive class, and $\{\text{NE}, \text{N}\}$ as negative class. In both datasets, four types of bond are included.

We compare our kernel with a pattern discovery-based method that uses frequent substructures as features of vector representations. Pattern discovery algorithms [11, 6] find all substructure patterns that appear more frequently than a given threshold in a dataset. In this paper, we use a frequent path finding algorithm [11] for constructing feature spaces. Suppose that the pattern discovery algorithm finds m frequent paths $\{path_1, path_2, \ldots, path_m\}$ in the dataset. The vector representation of a graph G is defined as

$$\begin{aligned} V_G^{num} &= (num(path_1, G), num(path_2, G), \ldots, \\ &\quad num(path_m, G)) && (20) \end{aligned}$$

where $num(path_i, G)$ is the number of times $path_i$ appears in graph G. The inner product of two vector representation $V_{G_1}^{num}$ and $V_{G_2}^{num}$ of two graphs G_1 and G_2 is defined as the following.

$$K^{num}(G_1, G_2) = \sum_{i=1}^{m} num(path_i, G_1) \cdot num(path_i, G_2) \quad (21)$$

Another possible vector representation uses a binary function $bin(path_i, G)$ which returns 1 when $path_i$ appears in

G, and returns 0 otherwise.

$$K^{bin}(G_1, G_2) = \sum_{i=1}^{m} bin(path_i, G_1) \cdot bin(path_i, G_2) \tag{22}$$

Note that our kernel also assumes the similar feature space, however, our kernel allows to use an edge more than once when checking whether a certain path appears in a graph. This implies that our kernel counts the appearances of paths approximately.

Computing our kernel needs to solve simultaneous linear equations (18) with a $|V_1| \cdot |V_2| \times |V_1| \cdot |V_2|$ matrix. However, the matrix is sparse since the number of non-zero elements is $|V_1| \cdot |V_2| \cdot \max_{v_1 \in V_1} |A(v_1)| \cdot \max_{v_2 \in V_2} |A(v_2)|$, and we can employ various kinds of efficient numerical algorithms. In this experiment, we just use an iterative method using the recursive equations (12).

As for the learning algorithm, for ease of implementation, we used the voted kernel perceptron [3] whose performance is known to be comparable to that of SVMs.

Table 1 - Table 4 show the test accuracy measured by leave-one-out cross validation. 'num' and 'bin' indicate the results for frequent-path-based kernels (21) and (22) respectively. 'MinSup' is a parameter for the pattern discovery algorithm that decides the minimum support. Although our graph kernel is not as well as the frequent-path-based kernels for mutag dataset, it is comparable to the frequent-path-based kernels for PTC dataset.

5 Conclusion

In this paper, we applied kernel methods to classification of graphs with vertex labels and edge labels. We defined a graph kernel for a pair of graphs by a random walk on a vertex product graph of the two graphs. Concretely, the kernel was defined to be the probability with which two label sequences generated by two synchronized random walks on the graphs were identical.

Next, we performed some experiments on predicting properties of chemical compounds to investigate how our kernel performed well on real data, and the results showed encouraging results.

Our kernel approximately counts all the appearances of the paths included in a graph, and at the same time, the whole process avoids NP-hard steps. This implies our kernel may be effective for larger graphs that pattern discovery algorithms suffer from, and we plan to apply our kernel to such datasets.

In this paper, we applied our kernel only to undirected graphs, however, we can naturally treat directed graphs such as semi-structured data and WWW structure by incorporating the edge directions into the edge labels. We plan to apply our kernel to such various datasets to investigate the ability of our kernel further.

References

[1] M. Collins and N. Duffy. Convolution kernel for natural language. In *Proc. of the 14th NIPS*, 2001.

[2] W. Frakes and R. Baeza-Yates.(eds.). *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, 1992.

[3] Y. Freund and R. Shapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3), 1999.

[4] D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of Calfornia in Santa Cruz, 1999.

[5] C. Helma, R. King, S. Kramer, and A. Srinivasan. The predictive toxicology challenge 2000-2001. *Bioinformatics*, 17(1):107–108, 2001.

[6] A. Inokuchi, T. Washio, and H. Motoda. An Apriori-based algorithm for mining frequent substructures from graph data. In *Proc. of the 4th PKDD*, pages 13–23, 2000.

[7] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1,2):95–114, 2000.

[8] J. Kandola, J. Shawe-Taylor, and N. Cristianini. On the application of diffusion kernel to text data. Technical report, Neurocolt, 2002. NeuroCOLT Technical Report NC-TR-02-122.

[9] H. Kashima and T. Koyanagi. Kernels for semi-structured data. In *Proc. of the 19th ICML*, 2002.

[10] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proc. of the 19th ICML*, 2002.

[11] S. Kramer and L. D. Raedt. Feture construction with version spaces for biochemical application. In *Proc. of the 18th ICML*, 2001.

[12] C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: a string kernel for SVM protein classification. In *Proc. of PSB 2002*, pages 564–575, 2002.

[13] T. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[14] A. Srinivasan, S. Muggleton, R. D. King, and M. Sternberg. Theories for mutagenicity: a study of first-order and feature based induction. *Artificial Intelligence*, 85(1-2):277–299, 1996.

[15] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[16] C. Watkins. Kernels from matching operations. Technical Report CSD-TR-98-07, University of London, Computer Science Department, Royal Holloway, 1999.

MinSup	bin	num
0.5 %	89.4%	88.3%
1 %	88.3%	87.8%
3 %	**89.9%**	**89.9%**
5 %	89.4%	86.2%
10 %	84.0%	84.6%
20 %	85.1%	83.5%

Table 1. Result for mutag (frequent-path-based kernel)

λ	Graph Kernel
0.1	78.7%
0.2	79.8%
0.3	81.9%
0.4	83.0%
0.5	83.5%
0.6	**85.1%**
0.7	**85.1%**
0.8	83.5%
0.9	84.e%

Table 2. Result for mutag (graph kernel)

	MM		FM		MR		FR	
MinSup	bin	num	bin	num	bin	num	bin	num
0.5%	61.0%	60.1%	57.3%	57.6%	59.0%	61.3%	63.8%	**66.7%**
1 %	59.8%	**61.0%**	59.0%	**61.0%**	59.3%	**62.8%**	**64.7%**	63.2%
3 %	59.2%	58.3%	59.6%	55.9%	57.8%	60.2%	63.2%	63.2%
5 %	56.8%	60.7%	58.2%	55.6%	55.5%	57.3%	64.1%	63.0%
10 %	57.4%	58.9%	**61.0%**	58.7%	58.4%	57.8%	60.1%	60.1%
20%	**61.6**%	**61.0%**	57.0%	55.3%	**60.2%**	56.1%	60.7%	61.3%

Table 3. Result for PTC (frequent-path-based kernel)

λ	MM	FM	MR	FR
0.1	62.8%	61.6%	**58.4%**	**66.1%**
0.2	63.4%	**63.4%**	54.9%	64.1%
0.3	63.1%	62.5%	54.1%	63.2%
0.4	62.8%	61.9%	54.4%	65.8%
0.5	64.0%	61.3%	56.1%	64.4%
0.6	**64.3%**	61.9%	56.1%	63.0%
0.7	64.0%	61.3%	56.7%	62.1%
0.8	62.2%	61.0%	57.0%	62.4%
0.9	62.2%	59.3%	57.0%	62.1%

Table 4. Result for PTC (graph kernel)

Active Mining from Hepatitis Data by Beam-wise GBI

Takashi Matsuda, Tetsuya Yoshida, Hiroshi Motoda and Takashi Washio
I.S.I.R., Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
{matsuda,yoshida,motoda,washio}@ar.sanken.osaka-u.ac.jp

Abstract

A machine learning technique called Graph-Based Induction (GBI) extracts typical patterns from graph data by stepwise pair expansion (pairwise chunking). Because of its greedy search strategy, it is very efficient but suffers from incompleteness of search. Improvement is made on its search capability without imposing much computational complexity by 1) incorporating a beam search, 2) using a different evaluation function to extract patterns that are more discriminatory than those simply occurring frequently, and 3) adopting canonical labeling to enumerate identical patterns accurately. This new algorithm, now called Beam-wise GBI, B-GBI for short, was applied to a real-world data, Hepatitis dataset provided by Chiba University. The spiral cycle of active mining was repeated three times to extract typical patterns from the dataset by B-GBI in close collaboration with a domain expert and examples of extracted patterns are reported. Our very preliminary results indicate that B-GBI can actually handle graphs with a few thousands nodes and extract discriminatory patterns.

1 Introduction

There have been quite a number of research work on data mining in seeking for better performance over the last few years. Better performance includes mining from structured data, which is a new challenge, and there has been little work on this subject. Since structure is represented by proper relations and a graph can easily represent relations, knowledge discovery from graph structured data poses a general problem for mining from structured data [6]. Some examples amenable to a graph mining are finding typical web browsing pattern, identifying typical substructure of chemical compounds, finding typical subsequences of DNA and discovering diagnostic rules from patient history records.

The majority of methods widely used are for data that does not have structure and is represented by attribute-value pairs. Decision tree[10, 11], and induction rules[8, 2] relate attribute values to target classes. Association rules often used in data mining also use this attribute-value pair representation. However, the attribute-value pair representation is not suitable for representing a more general data structure, and there are problems that need a more powerful representation. Most powerful representations that can handle relation and thus, structure, would be inductive logic programming (ILP) [9] which uses the first-order predicate logic. It can represent general relationships embedded in data, and has a merit that domain knowledge and acquired knowledge can be utilized as background knowledge. However, in exchange for its rich expressibility, the time complexity causes problem [3].

It is widely advocated that knowledge discovery from databases (KDD) is a recurring process which consists of data preprocessing, mining, evaluation, etc [4]. Active mining promotes this idea further by incorporating the active involvement of humans (e.g., users and domain experts) not only in mining but also in information gathering and evaluation of extracted knowledge. Especially, evaluations and feedbacks for the extracted knowledge from humans are intensively enforced both for improving a mining method and for determining and gathering necessary information in the next cycle of KDD process. Thus, active mining follows a spiral model of scientific discovery in spirit to extract useful knowledge from data by repeating the cycle of active information gathering, user-centered active mining and active user reaction, as shown in Figure 1. This paper reports preliminary results of active mining from a real-world data, Hepatitis dataset that was provided by Chiba University, using the improved Graph-Based Induction (GBI) [13, 7].

GBI is a technique which was devised for discovering typical patterns in a general graph data by recursively chunking two adjoining nodes. It can handle a graph data having loops (including self-loops) with colored/uncolored nodes and links. There can be more than one link between any two nodes. GBI is very efficient because of its greedy search. GBI can use various evaluation functions based on frequency. It is not, however, suitable for pattern extraction

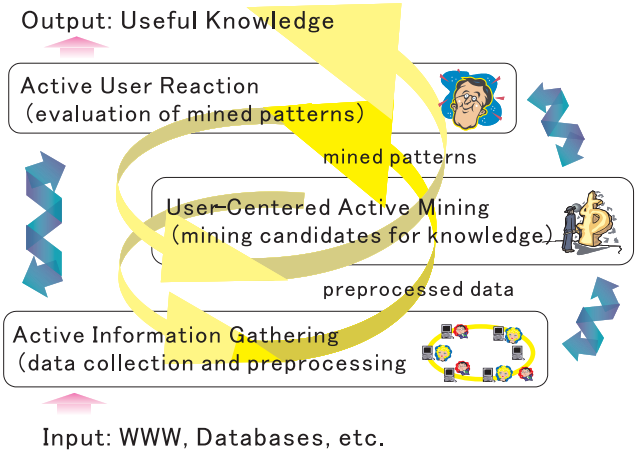


Figure 1. A spiral cycle in active mining

from a graph structured data where many nodes share the same label because of its greedy recursive chunking without backtracking, but still effective in extracting patterns from such graph structured data where each node has a distinct label (*e.g.*, World Wide Web browsing data) or where some typical structures exist even if some nodes share the same labels (*e.g.*, chemical structure data containing benzene rings etc) [7].

Efficiency of GBI comes from its greedy search in exchange for search incompleteness. It cannot find all the important typical patterns although our past application to various domains produced acceptable results [7]. In this paper we first report the improvement made to enhance the search capability without sacrificing efficiency too much by 1) incorporating a beam search, 2) using a different evaluation function to extract patterns that are more discriminatory than those simply occurring frequently, and 3) adopting canonical labeling to enumerate identical patterns accurately. This new algorithm is implemented and now called Beam-wise GBI, B-GBI for short. After that, we report on an initial result of the analysis in which B-GBI was applied to the Hepatitis dataset whose temporal records was converted into graph structured data with respect to time correlation. The spiral cycles of active mining was repeated three times to extract patterns by B-GBI in close collaboration with a domain expert. Feedbacks from the expert were used to replace redundant attributes (examinations) with a newly defined one, add new attributes, remove some others by feature selection, and modify the graph structure. Preliminary results indicate that B-GBI can actually handle graphs with thousands of nodes and extract discriminatory patterns.

The paper is organized as follows. Section 2 describes the framework of B-GBI focusing on the improvement made to GBI. Section 3 reports the preliminary results applied to the hepatitis dataset. Section 4 concludes the paper with summary of the results and the planned future work.

2 Beam-wise Graph-Based Induction

GBI employs the idea of extracting typical patterns by stepwise pair expansion, as shown in Figure 2. "Typicality" is characterized by the pattern's frequency or the value of some evaluation function of its frequency. It is possible to extract typical patterns of various sizes by repeating the stepwise pair expansion (pairwise chunking). Note that the search is greedy. No backtracking is made. This means that in enumerating pairs no pattern which has been chunked into one node is restored to the original pattern. Because of this, all the "typical patterns" that exist in the input graph are not necessarily extracted. The problem of extracting all the isomorphic subgraphs is known to be NP-complete. Thus, GBI aims at extracting only meaningful typical patterns. Its objective is not finding all the typical patterns nor finding all the frequent patterns.

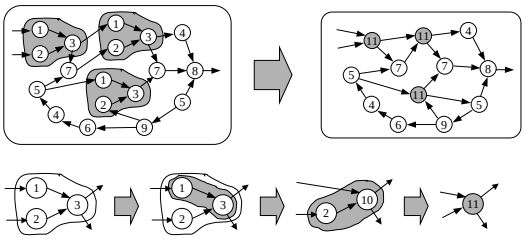


Figure 2. The basic idea of the GBI method

For finding a pattern that is of interest, any of its subpatterns must be of interest because of the nature of repeated chunking. Frequency measure satisfies this monotonicity. However, if the criterion chosen does not satisfy this monotonicity, repeated chunking may not find good patterns, even though the best pair based on the criterion is selected at each iteration. This motivated us to improve GBI allowing to use two criteria, one for frequency measure for chunking and the other for finding discriminatory patterns after chunking. The latter criterion does not necessarily hold monotonicity property. Any function that is discriminatory can be used, such as Information Gain [10], Gain Ratio [11] and Gini Index [1], all of which are based on frequency.

When each node has a distinct label in the input graph, no ambiguity arises in selecting a pair to be chunked and GBI performs well. However, since the search in GBI is greedy, when the same label is shared by plural nodes in the input graph, there arises ambiguity when there are ties in the frequency or there is a chain of nodes of the same label. For example, in the case of the structure like $a \rightarrow a \rightarrow a$, we don't know which $a \rightarrow a$ is best to chunk.

To enhance the search capability, a beam search is incorporated to GBI within the framework of greedy search. A certain fixed number of pairs ranked from the top are allowed to be chunked in parallel. To prevent each branch from growing exponentially, the total number of pairs to chunk is fixed to a pre-specified value at each level of branch. Thus, at any iteration step, there is always a fixed number of chunking that is performed in parallel.

The new stepwise pair expansion repeats the following four steps.

Step 1 Extract all the pairs consisting of connected two nodes in all the graphs.

Step 2a Select all the typical pairs based on the criterion from among the pairs extracted in Step 1, rank them according to the criterion and register them as typical patterns. If either or both nodes of the selected pairs have already been rewritten (chunked), they are restored to the original patterns before registration.

Step 2b Select, from among the pairs extracted in Step 1, a fixed number of frequent pairs from the top and register them as the patterns to chunk. If either or both nodes of the selected pairs have already been rewritten (chunked), they are restored to the original patterns before registration. Stop when there is no more pattern to chunk.

Step 3 Replace each of the selected pairs in Step 2b with one node and assign a new label to it. Delete a graph for which no pair is selected and branch (copy) a graph for which more than one pair are selected. Rewrite each remaining graph by replacing all the occurrence of the selected pair in the graph with a node with the newly assigned label. Go back to Step 1.

The output of the B-GBI is a set of ranked typical patterns extracted at Step 2a. These patterns are typical in the sense that they are more discriminatory than non-selected patterns in terms of the criterion used.

Another improvement made in conjunction with B-GBI is canonical labeling. GBI assigns a new label for each newly chunked pair. Because it recursively chunks pairs, it happens that the new pairs that have different labels happen to be the same pattern (subgraph). A simple example is shown in Figure 3.

To identify whether the two pairs represent the same pattern or not, each pair is represented by its canonical label[12, 5] and only when the label is the same, they are regarded as identical. The basic procedure of canonical labeling is as follows. Nodes in the graph are grouped according to their labels (node colors) and the degrees of node (number of links attached to the node) and ordered lexicographically. Then an adjacency matrix is created using this

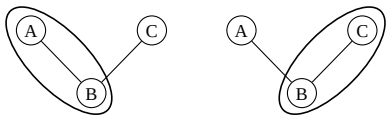


Figure 3. Two Different Pairs Representing Identical Pattern

node ordering. When the graph is symmetric, the upper triangular elements are concatenated scanning either horizontally or vertically to codify the graph. When the graph is asymmetric, all the elements in both triangles are used to codify the graph in a similar way. If there are more than one node that have identical node label and identical degrees of node, the ordering which results in the maximum (or minimum) value of the code is searched. The corresponding code is the canonical label. Let M be the number of nodes in a graph, N be the number of groups of the nodes, and $p_i (i = 1, 2, \ldots, N)$ be the number of the nodes within group i. The search space can be reduced to $\prod_{i=1}^{N}(p_i!)$ from $M!$ by using canonical labeling. The code of an adjacency matrix for the case in which elements in the upper triangle are vertically concatenated is defined as

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ & a_{22} & \ldots & a_{2n} \\ & & \ddots & \vdots \\ & & & a_{nn} \end{pmatrix}$$

$$code(A) = a_{11}a_{12}a_{22}a_{13}a_{23}\ldots a_{nn} \quad (1)$$

$$= \sum_{j=1}^{n}\sum_{i=1}^{j}\left((L+1)^{\{\sum_{k=j+1}^{n} k\}+j-i}a_{ij}\right) \quad (2)$$

Here L is the number of different link labels. It is possible to further prune the search space. We choose the option of vertical concatenation. Elements of the adjacency matrix of higher ranked nodes form higher elements of the code. Thus, once the locations of higher ranked nodes in the adjacency matrix are fixed, corresponding higher elements of the code are also fixed and are not affected by the order of elements of lower ranks. For example, in Eq. 1 elements that the first two ranked nodes can decide are the first 3 elements in the $code(A)$ and no elements corresponding to the nodes of the lower ranks are included. This reduces the search space of $\prod_{i=1}^{N}(p_i!)$ to $\sum_{i=1}^{N}(p_i!)$.

However, there is still a problem of combinatorial explosion for a case where there are many nodes of the same labels and the same degrees of node such as the case of chemical compounds because the value of p_i becomes large. What

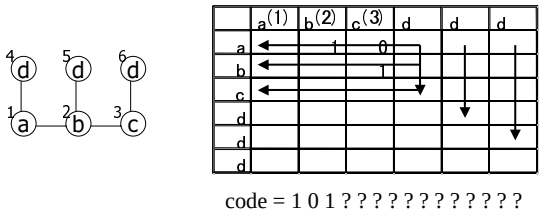


Figure 4. Determination of Node Ordering within a Group

we can do is to make the best of already determined nodes of higher ranks. Assume that the nodes $v_i \in V(G)(i = 1, 2, \ldots, N)$ are already determined in a graph G. Consider finding the order of the nodes $u_i \in V(G)(i = 1, 2, \ldots, k)$ of the same group that gives the maximum code value. The node that comes to v_{N+1} is the one in $u_i(i = 1, \ldots, k)$ that has a link to the node v_1 because the highest element that v_{N+1} can make is a_{1N+1} and the node that makes this element non 0, that is, the node that is linked to v_1 gives the maximum code. If there are more than one node or no node at all that has a link to v_{N+1}, the one that has a link to v_2 comes to v_{N+1}. Repeating this process determines which node comes to v_{N+1}. If no node can't be determined after the last comparison at v_N, permutation within the group is needed. Thus, the computational complexity in the worst case is still exponential.

This is explained using an example in Figure 4. Assume that nodes 1, 2 and 3 have already been determined. Nodes 4, 5 and 6 are in the same group. The fourth node is the node that has a link to the highest ranked node 1, which is the node 4. Likewise, the fifth and the sixth nodes are the nodes 5 and 6 respectively. In this case, node ordering is uniquely determined. If l nodes can be determined by this procedure, the search space can be reduced from $k!$ to $(k - l)!$.

3 Extracting Patterns from Hepatitis Dataset

We have attempted to analyze the hepatitis dataset that was provided by Chiba University. The results shown in this paper is very preliminary. The dataset contains long time-series data (about 20 years from 1882 to 2001) on laboratory examinations of 771 patients of hepatitis B and C. The data can be broadly split into two categories. The first data include administrative information such as patient's information (age and date of birth), pathological classification of the disease, date of biopsy, result of biopsy, and duration of interferon therapy. The second data include temporal records of blood test and urinalysis. It can be further split into two subcategories, in-hospital and out-hospital examination data. In-hospital examination data contain the results of 230 examinations that were performed using the hospital's equipment. Out-hospital examination data contain the results of 753 examinations, including comments of staffs, performed using special equipment on outside facilities. Consequently, the temporal data contain the results of 983 types of examinations. These were given in 6 different tables.

3.1 Preprocessing and Data Conversion to Graphs

The original data provided are averaged for a specified interval and a new reduced dataset is generated because the date of visit is not synchronized across different patients and it is considered that the progress of hepatitis is slow. Numerical average is taken for numeric attributes and maximum frequent value is used for nominal attributes over the interval. Further numeric values are discretized when the normal range is given. If there are no data in the interval, these are treated as missing values and no attempt is made to estimate these values.

One patient record is mapped into one colored directed graph. Assumption is made that there is no direct correlation between two sets of measurements that are more than a predefined interval (e.g., two years). Thus, time correlation is considered only within the interval. Figure 5 shows an example of graph for a particular patient when the activity is taken as class and the interval for taking an average is set to 1 month. In this figure a star-shaped subgraph represents values of a set of medical examination for the interval. The center of the subgraph is a hypothetical node for the interval. An edge from the hypothetical node represents an examination and the node connected to the edge represents the value of the examination. When the value of an examination is missing for an interval, there is no corresponding node and edge in the subgraph. The hypothetical node of a subgraph is connected to every hypothetical node up to the specified interval. Figure 5 represents that the patient did not received interferon therapy since the label of the node connected to the edge "ifn" (which represents whether the patient has received interferon therapy or not) is "n" in all subgraphs.

Equation 3 was used as the typicality evaluation function. Here, n_{C_k} indicates the number of instances with class C_k that have a typical pattern in question and N_{C_k} is the total number of instances with class C_k. This function takes the value $\frac{1}{K}$ when the pattern is irrelevant and the class distribution does not change.

$$Max. \left\{ \frac{\frac{n_{C_1}}{N_{C_1}}}{\sum_{k=1}^{k=K} \frac{n_{C_k}}{N_{C_k}}}, \frac{\frac{n_{C_2}}{N_{C_2}}}{\sum_{k=1}^{k=K} \frac{n_{C_k}}{N_{C_k}}}, ..., \frac{\frac{n_{C_k}}{N_{C_k}}}{\sum_{k=1}^{k=K} \frac{n_{C_k}}{N_{C_k}}}, \right\} \quad (3)$$

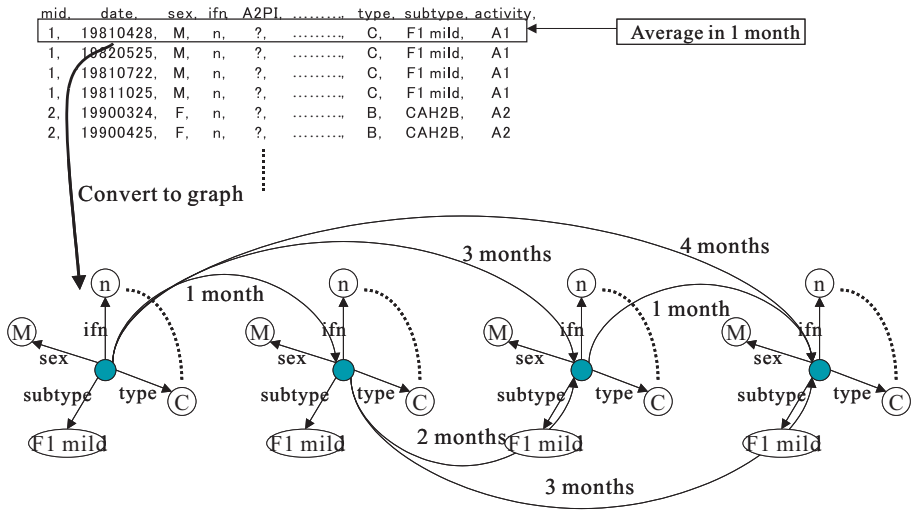


Figure 5. An example of graph when the activity is taken as a class

Table 1. Size of the graphs (class = Activity)

Class	A1	A2	A3	All
No. of graphs	51	54	11	116
No. of average nodes	1975	1758	1776	1855
Max. node number	6723	6688	4572	6723
Min. node number	63	53	214	53

Table 2. Number of extracted patterns (class = Activity)

Threshold	No. of patterns
0.4	192826
0.5	172437
0.6	140991
0.7	117628
0.8	105426

3.2 Extracted Patterns for Biopsy

Our first attempt focuses on finding typical data patterns, if any, that are strongly correlated to fibrosis (progress of fibrosis, discrete values: F0(mild)-F4(severe)) and activity (activity of virus, discrete values: A1(mild)-A3(severe)). The biopsy data are not measured for all patients and limiting to non acute hepatitis B resulted in 116 patients. The duration of interval for taking average was set to 28 days. After the data were averaged, numeric values were discretized into three intervals (low, normal and high) when the normal range is given. The duration for time correlation was set to two years. The beam width was set at 3. Table 1 and 2 show the size of generated graphs and the number of extracted patterns for each threshold. The maximum graph size is over 5,000 (number of nodes). Huge number of patterns (more than 100,000) is extracted. The threshold 1/No_of_Class indicates no discriminatory power (meaning the same as the default distribution). Thus threshold = 0.8 indicates that the extracted patterns are highly discriminatory. The computation time for threshold = 0.4 is 16 hours by PC with CPU of Athlon MP 1600+ and Main memory of 1GB.

A few top ranked patterns are shown for each class in Figs. 6 and 7. It turns out that many of the highly discriminatory rules consist of patterns of one shot measurements, which is contrary to our expectation. We show here only two of them. However, as the number of patients in each class whose measurements satisfy these patterns indicates, these patterns are highly discriminatory (See the initial distribution in Table 1 for class=activity. The initial distribution for class=fibrosis is F0=1,F1=49, F2=33, F3=26, F4=12 (We deleted F0 from the analysis because there is only one patient)). These results were shown to the medical doctor who provided us with the data. He was not used to see the measurements this way and had difficulty in interpreting the patterns, but most of the patterns were interpretable and there was no surprise.

3.3 Extracted Patterns for Hepatitis B Virus

Analysis of the dataset with the domain expert revealed that the original dataset includes many redundant attributes (examinations). To remove redundant attributes, he de-

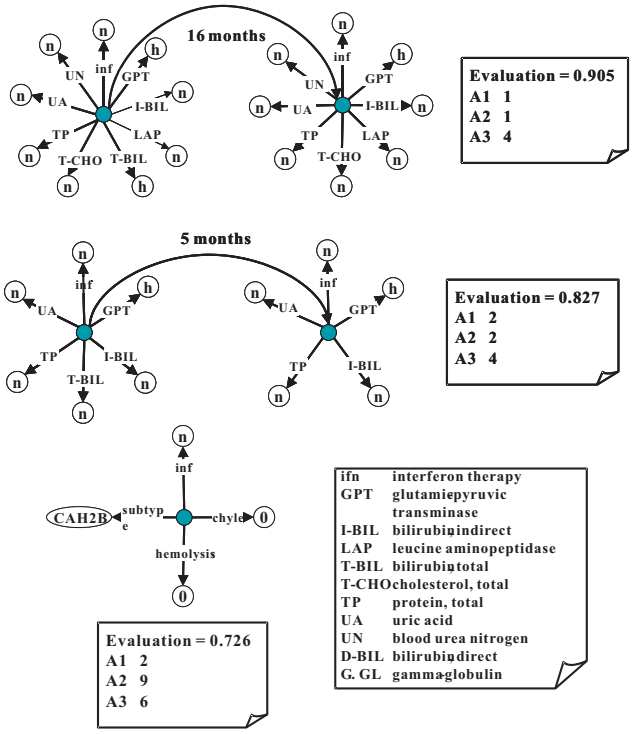


Figure 6. Example of extracted patters (class=activity)

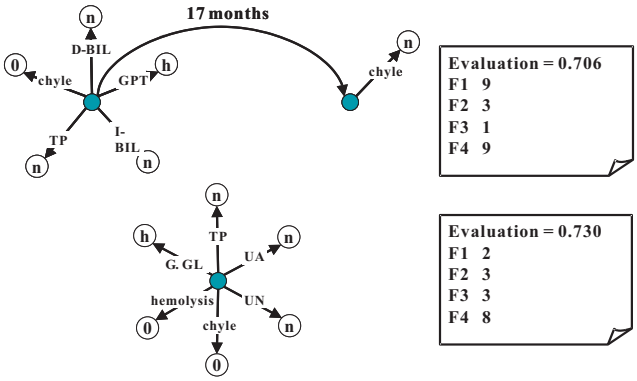


Figure 7. Example of extracted patters (class=fibrosis)

fined a new rule to determine the status of hepatitis B virus (HBV), as shown in Table 3. The newly defined attribute (HBV) was added to the dataset and the redundant ones were removed.

The expert analyzed the extracted patterns in Subsection 3.2 and suggested to investigate the changes of measurements over 6 months since the clinical condition of hepatitis progresses slowly. Based on this suggestion, the data

Table 3. Status of hepatitis B virus (HBV)

HBS-AG	HBE-AG	HBE-AB	HBS-AB	HBV
+	+	-	-	active
+	-	+	-	inactive
-	-	+	+	cured

Table 4. Size of the graphs (class = HBV)

Class	active	inactive	cured
No. of graphs	24	49	25
No. of average nodes	351	474	409
Max. node number	703	788	717
Min. node number	17	32	135

were averaged over 6 months in data preprocessing. In addition, the expert suggested that the short-term change of clinical condition is often reflected on the deviation of values of GOT, GPT, TTT, ZTT. Thus, the standard deviations of these examinations for 6 months were taken and added as new attributes. These attributes were then discretized into five values (in GOT, GPT) and three values (in TTT, ZTT) based on their histograms, respectively. Furthermore, we carried out feature selection with the expert to reduce the number of attributes to 23. Finally, the expert indicated that the symbol at the end of value (e.g., "H" and "L" in the value 23.6H and 40.5L) was more significant than the numerical value itself (e.g., 23.6 and 40.5). Based on this comment, numerical values were discretized as follows. If a symbol is attached at the end of value, the symbol is taken as the discretized value; if not, "n" which means normal, is taken as the value. Next, discretized values were averaged over 6 months by taking the maximum frequent value of 6 months. The preprocessed data were converted into graph structured data by spanning the links for time correlation up to 10 years.

The added attribute HBV is treated as class and patterns were extracted by G-GBI. The values of HBS-AG, HBE-AG, HBE-AB and HBS-AB does not necessarily conform to the rule in Table 3 and limiting to the data with the status of HBV resulted in 98 patients. The beam width was set at 3. Table 4 shows the size of generated graphs, which is greatly reduced compared with that in Table 1. The computation time was 2.2 hours (CPU of Athlon MP 1600+, Main memory of 3GB). Table 5 shows the number of extracted patterns for each threshold.

Some of extracted patterns are shown in Figure 8. The expert commented that the top pattern was hardly understandable from the domain knowledge since it indicates the correlation of measurements between 9.5 years. The middle

Table 5. Number of extracted patterns (class = HBV)

Threshold	No. of patterns
0.4	67239
0.5	58868
0.6	48849
0.7	39579
0.8	37743

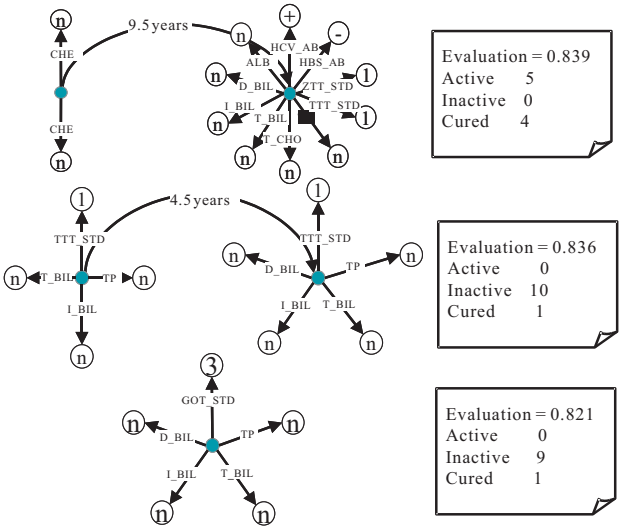


Figure 8. Example of extracted patters (class=HBV, up to 10 years)

pattern was evaluated as reasonable since it accords with the known medical knowledge that the change of TTT is small when the virus is inactive. The bottom pattern was surprising for the expert since it indicates that the virus can be inactive even if the change of GOT is large.

3.4 Extracted Patterns for HBV by Limiting Time Correlation

Based on the comment from the expert that patterns with too long duration between measurements are hardly understandable, another graph-structured data were created for the data in Subsection 3.3 by limiting time correlation up to two years. B-GBI was then applied to the data by treating HBV as class. Examples of extracted patters are shown in Figure 9. Table 6 shows the number of extracted patterns for each threshold, which is reduced compared with Table 5 by limiting time correlation up to two years.

Table 6. Number of extracted patterns (class = HBV, up to 2 year)

Threshold	No. of patterns
0.4	17505
0.5	14641
0.6	11567
0.7	9016
0.8	8512

The top pattern in Figure 9 is typical for HBV = active and the expert evaluated it as reasonable since it indicates that the standard deviations of GOT and TTT are 2 when HBV is active. The middle pattern is typical for HBV = inactive and was also evaluated as reasonable since the standard deviations of GOT,GPT,TTT are 1 in that pattern. The bottom patterns is typical for HBV = cured, and again, it was evaluated as reasonable for cured patients since the standard deviations of TTT and ZTT are 1.

The expert commented that removing redundant examinations and incorporating fluctuation of the numeric values helped extract more reasonable patterns in terms of the domain knowledge. This was made possible through the intensive interaction with the expert during the three cycles of analysis. Especially, he showed interest in the extracted patterns which indicated the correlation of the standard deviation of values such as TTT and GPT.

4 Conclusion

Graph based induction GBI is improved in three aspects by incorporating: 1) two criteria, one for chunking and the other for task specific criterion to extract more discriminatory patterns, 2) beam search to enhance search capability and 3) canonical labeling to accurately count identical patterns. The improved B-GBI was applied to a real-world hepatitis data to search for measurement patterns that are strongly correlated to the status of liver and virus activity. The spiral cycle of active mining using B-GBI was repeated three times in collaboration with a domain expert and some of the extracted patterns were confirmed as reasonable by the expert. Each time a comment of the expert is obtained, this is reflected to the analysis of the next cycle. *e.g.*, the dataset itself is modified and/or how it is mapped to the graph structure is modified, which helped improve the quality of the extracted patterns. Although the results are still preliminary to discuss the values of the discovered patterns, we believe that B-GBI can actually handle graphs with a few thousands nodes and extract discriminatory patterns.

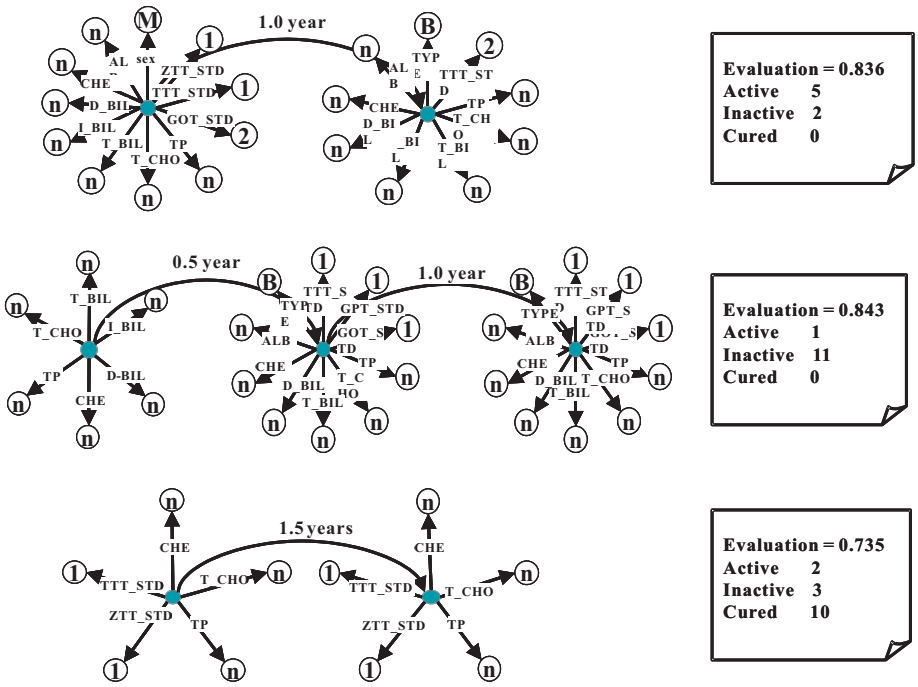


Figure 9. Example of extracted patters (class=HBV, up to 2 years)

Immediate future work includes to 1) seek the appropriate duration of time correlation when converting to graph-structured data 2) continue the analysis of the hepatitis B virus data by discretizing the measurements in more reasonable way at preprocessing and 3) extract typical patterns for hepatitis C to discriminate the effect of interferon therapy.

Acknowledgement

This work was partially supported by the grant-in-aid for scientific research on priority area “Active Mining” (No. 13131101, No. 13131206) funded by the Japanese Ministry of Education, Culture, Sport, Science and Technology. Special thanks are due to Chiba University for providing us with the hepatitis dataset.

References

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. The cn2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

[3] L. Dehaspe, H. Toivonen, and R. D. King. Finding frequent substructures in chemical compound. In *Proc. the 4th International conference on Knowledge Discovery and Data Mining*, pages 30–36, 1998.

[4] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From Data Mining to Knowledge Discovery: An Overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34. AAAI Press, 1996.

[5] S. Fortin. The graph isomorphism problem, 1996.

[6] T. Matsuda, , T. Motoda, H. Yoshida, and T. Washio. Knowledge discovery from structured data by beam-wise graph-based induction. In *Proc. of the 7th Pacific Rim International Conference on Artificial Intelligence*, pages 255–264, 2002.

[7] T. Matsuda, T. Horiuchi, H. Motoda, and T. Washio. Extension of graph-based induction for general graph structured data. In *Knowledge Discovery and Data Mining: Current Issues and New Applications, Springer Verlag, LNAI 1805*, pages 420–431, 2000.

[8] R. S. Michalski. Learning flexible concepts: Fundamental ideas and a method based on two-tiered representaion. *In Machine Learning, An Artificial Intelligence Approiach*, 3:63–102, 1990.

[9] S. Muggleton and L. de Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19(20):629–679, 1994.

[10] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[11] J. R. Quinlan. *C4.5:Programs For Machine Learning*. Morgan Kaufmann Publishers, 1993.

[12] R. C. Read and D. G. Corneil. The graph isomorphism disease. *Journal of Graph Theory*, 1:339–363, 1977.

[13] K. Yoshida and H. Motoda. Clip : Concept learning from inference pattern. *Journal of Artificial Intelligence*, 75(1):63–92, 1995.

Prototyping Medical Test Results in Chronic Hepatitis Data with the EM Algorithm on Multi-Dice Models

Takeshi Watanabe Einoshin Suzuki
Electrical and Computer Engineering, Yokohama National University,
79-5, Tokiwadai, Hodogaya, Yokohama, 240-8501, Japan
nabekun@slab.dnj.ynu.ac.jp suzuki@ynu.ac.jp

Hideto Yokoi Katsuhiko Takabayashi
Division for Medical Informatics, Chiba-University Hospital,
1-8-1 Inohana, Chuo, Chiba, 260-8677, Japan
yokoih@telemed.ho.chiba-u.ac.jp takaba@ho.chiba-u.ac.jp

Abstract

This paper presents an example of active mining endeavor in which we estimate prototypes of medical test results in chronic hepatitis data with the EM algorithm. As a result of trials and errors with medical domain experts, we have come to invent a multi-dice model, which represents a probabilistic model of the prototype. Experiments show that most of the obtained prototypes can be interpreted easily and clearly in the medical context, and our proposed method is promising in both recognition and understanding of a patient.

1 Introduction

A typical medical test result consists of a time stamp, a patient's identification number (ID), and a value of a medical test. Typically, the number of medical tests which a patient undergoes in a day is much smaller than the total number of medical tests. We initially thought that a collection of medical test results could be handled as a transactional data set, which is frequently used in association-rule mining [1]. However, we have come to conclude that special care is required in discriminating a missing value from a normal value as well as selection and pre-processing of the data set. Chronic hepatitis data [2], which is the target of active mining in this paper, contains a large number of medical test results, and cannot be effectively analyzed with a conventional learning/discovery method.

As a first step to circumvent this problem, we, computer scientists and medical experts, have tried to prototype test values by the EM algorithm for mixture probabilistic models [4]. We show that the prototypes are useful in understanding and cleaning the data on a patient level, which is highly appreciated by the medical experts. This paper mainly shows the process of trials and errors, and contribution to the medical domain of the current results.

2 Estimating Probabilistic Mixture Models

2.1 Definition of the Problem

We regard a set of values of medical tests which a patient undergoes in a specific time of a day[1] an instance. Let A be a set of possible medical tests $A = \{a_1, a_2, \cdots, a_{n(A)}\}$, where a_l and $n(A)$ represent a medical test and the number of medical tests respectively. The input to the problem is a data set X, which consists of $n(X)$ instances $X = \{\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_{n(X)}\}$. Here x_i represents a value vector which consists of $n(A)$ values $\mathrm{x}_i = \{v_i(a_1), v_i(a_2), \cdots, v_i(a_{n(A)})\}$, where $v_i(a_l)$ represents a value of a medical test a_l for an instance x_i. The output of the problem is a set K which consists of $n(K)$ prototypes $K = \{k_1, k_2, \cdots, k_{n(K)}\}$. The probability that an instance x_i occurs can be given by

$$p(\mathrm{x}_i) = \sum_{j=1}^{n(K)} p(\mathrm{x}_i|k_j)p(k_j), \quad (1)$$

where $p(k_j)$ represents the probability that a prototype k_j occurs, and $p(\mathrm{x}_i|k_j)$ represents the conditional probability that x_i occurs given k_j.

[1] In the data set[2], a set of transactions which have the same patient ID, the day, and the time

2.2 EM Algorithm

This paper employs the EM algorithm [4] for estimating probabilistic mixture models. This algorithm gives the maximum likelihood estimates of $p(k_j)$ and $p(\mathrm{x}_i|k_j)$ with hill-climbing search [7]. These estimates can be defined as the values which minimize the negative log-likelihood ε.

$$\varepsilon = -\sum_{i=1}^{n(X)} \ln \left(\sum_{j=1}^{n(K)} p(\mathrm{x}_i|k_j)p(k_j) \right) \tag{2}$$

An intuitive representation of the EM algorithm is as follows.

1. Give initial values for $p(k_j)$ and $p(\mathrm{x}_i|k_j)$.
2. Calculate the value of $p(k_j|\mathrm{x}_i)$ with (1) and the Bayes rule.

$$p(k_j|\mathrm{x}_i) = \frac{p(\mathrm{x}_i|k_j)p(k_j)}{p(\mathrm{x}_i)} \tag{3}$$

   The procedure for obtaining $p(\mathrm{x}_i|k_j)$ depends on the probabilistic model, and will be explained in each model in section 3.

3. Update the values of $p(k_j)$ and $p(\mathrm{x}_i|k_j)$.

$$p^{\mathrm{new}}(k_j) = \frac{1}{n(X)} \sum_{i=1}^{n(X)} p(k_j|\mathrm{x}_i) \tag{4}$$

   The procedure for obtaining $p^{\mathrm{new}}(\mathrm{x}_i|k_j)$ depends on the probabilistic model, and will be explained in each model in section 3.

4. Iterate step 2 and step 3 until convergence.

3 Proposed Probabilistic Models

3.1 Single-Dice Model

A typical medical test is assigned a range for normal values, and a value which resides out of this range is judged as abnormal. Assume we assign 1 and 0 to an abnormal value and any other value respectively, then medical test results can be transformed to a transactional data set since most of test values are 0 and the data set represents a sparse binary table.

In this section, a prototype is represented by a dice with $n(A)$ faces, and we call this prototype a single-dice model. Recall that $n(A)$ represents the number of medical tests. The probability that a face of a dice occurs represents the probability that the value of the corresponding medical test becomes abnormal. A multinomial distribution [5] models the numbers of occurrence of exhaustive and mutually exclusive events by a series of independent trials, e.g. the probabilities of faces in throwing a dice. We believed that a multinomial distribution was effective in analyzing any kinds of transactional data sets since it has been successfully employed in profiling of purchase behavior [3]. In order to prototype abnormal medical test values with the single-dice model, we classify a value $v_i(a_l)$ of an instance x_i 1 (abnormal) and 0 (any other value).

$$v_i(a_l) = \begin{cases} 1 & (\text{abnormal value}) \\ 0 & (\text{any other value}) \end{cases} \tag{5}$$

A prototype k_j consists of probabilities that values of medical tests become abnormal.

$$k_j = \{p_j(a_1), p_j(a_2), \cdots, p_j(a_{n(A)})\} \tag{6}$$

where $p_j(a_l)$ represents the probability that a medical test a_l becomes abnormal in the prototype k_j. From the definition of a multinomial distribution [5], we defined the conditional probability $p(\mathrm{x}_i|k_j)$ in (3) as follows[2].

$$p(\mathrm{x}_i|k_j) = \frac{n(A)!}{\prod_{l=1}^{n(A)} v_i(a_l)!} \prod_{l=1}^{n(A)} p_j(a_l)^{v_i(a_l)} \tag{7}$$

The conditional probability $p(\mathrm{x}_i|k_j)$ is updated using

$$p_j^{\mathrm{new}}(a_l) = \frac{\sum_{i=1}^{n(X)} p(k_j|\mathrm{x}_i)v_i(a_l)}{\sum_{i=1}^{n(X)} p(k_j|\mathrm{x}_i) \sum_{l=1}^{n(A)} v_i(a_l)} \tag{8}$$

It turned out by experiments that the single-dice model is inappropriate for the chronic hepatitis data. The model suffers from three weak points: 1) classification of medical test values to normal and abnormal is coarse, 2) it regards a normal value and a missing value equivalently, and 3) it neglects the fact that a transaction contains one value for a medical test in this data set.

3.2 Multi-Coin Model and Multi-Dice Model

In order to cope with the first problem in the last section, we discretize values of a medical test a_l as follows $R(a_l) = \{r_1(a_l), r_2(a_l), \cdots, r_{n(a_l)}(a_l)\}$, where $n(a_l)$ and $r_m(a_l)$ represent the number of labels in the discretization and the m-th label. For the second problem, we discriminate a missing value from others. Let $\mathrm{x}_i = \{v_i(a_1), v_i(a_2), \cdots, v_i(a_{n(A)})\}$, then $v_i(a_l)$ can take one of

[2] Note that we committed several errors here. For instance, $n(A)$ should be replaced by the number of abnormal test values in x_i

the values in the right-hand side.

$$v_i(a_l) = \begin{cases} r_1(a_l) \\ r_2(a_l) \\ \vdots \\ r_{n(a_l)}(a_l) \\ - \quad \text{(untested)} \end{cases} \tag{9}$$

For the third problem, we model each medical test as a dice throwing, thus we need $n(A)$ dices to model $n(A)$ medical tests. Note that (9) represents a dice throwing where one has a choice of not throwing it.

A prototype k_j is represented as

$$k_j = (k_{j1}, k_{j2}, \cdots, k_{jn(A)}) \tag{10}$$

$$\text{where } k_{jl} = (p_j(r_1(a_l)),\ p_j(r_2(a_l)), \cdots p_j(r_{n(a_l)}(a_l))), \tag{11}$$

where $p_j(r_m(a_l))$ represents the probability that a value, not including an untested value, of a medical test a_l becomes $r_m(a_l)$ in k_j. We call this prototype a multi-dice model.

In (3), the conditional probability $p(\mathrm{x}_i|k_j)$ is given by

$$p(\mathrm{x}_i|k_j) = \prod_{l=1}^{n(A)} p_j(v_i(a_l)), \tag{12}$$

where we define $p_j(v_i(a_l)) = 1$ if $v_i(a_l) = -$.

The conditional probability $p(\mathrm{x}_i)$ is updated using

$$p_j^{new}(r_m(a_l)) = \frac{\sum_{i=1}^{n(X)} p(k_j|\mathrm{x}_i)p(k_j)\gamma_1(\mathrm{x}_i, k_j, l, m)}{\sum_{i=1}^{n(X)} p(k_j|\mathrm{x}_i)p(k_j)\gamma_2(\mathrm{x}_i, k_j, l, m)} \tag{13}$$

where

$$\gamma_1(\mathrm{x}_i, k_j, l, m) = \begin{cases} 1 & (v_i(a_l) = r_m(a_l)) \\ 0 & (v_i(a_l) \neq r_m(a_l)) \end{cases} \tag{14}$$

$$\gamma_2(\mathrm{x}_i, k_j, l, m) = \begin{cases} 1 & (v_i(a_l) \neq -) \\ 0 & (v_i(a_l) = -) \end{cases} \tag{15}$$

Note that the multi-dice model degenerates to a "multi-coin" model if we discretize a medical test value to normal and abnormal, i.e. $\forall l\ n(a_l) = 2$. In fact, after the failure of the single-dice model, we first invented the multi-coin model, then have come to consider the multi-dice model, i.e. $n(a_l) > 2$, after the trials and errors in section 4.2.

4 Experiment

4.1 Single-Dice Model

4.1.1 Conditions

We employ chronic hepatitis data [2], which was donated by Chiba-University Hospital. This data set consists of 58,716 instances, and the number of medical tests is 458. The single-dice model, which was described in section 3.1, was first employed as a representation of a prototype. Recall that such a prototype represents a pattern on abnormal values.

The EM algorithm, due to its use of hill-climbing search, does not necessarily converge to a global optimum solution. To cope with this problem, we applied the algorithm 100 times with randomly-selected initial values, and returned the solution with the smallest negative log-likelihood ε in (2). We varied the number $n(K)$ of prototypes $2, 3, \cdots, 10$, and terminated the iteration either when each value of parameters changes less than 0.01% or the number of iterations is 100.

We also tried to cluster obtained prototypes based on similarities. The similarity $\beta(k, l)$ of a pair of prototypes k and l is based on divergence $D(k\|l)$, which measures the distance between two probabilistic distributions.

$$\beta(k, l) = \frac{D(k\|l) + D(l\|k)}{2} \tag{16}$$

$$D(k\|l) = \sum_{i=1}^{c} p_{li} \ln \frac{p_{li}}{p_{ki}} \tag{17}$$

where c represents the number of events in k and l, and $k = (p_{k1}, p_{k2}, \cdots, p_{kc})$, $l = (p_{l1}, p_{l2}, \cdots, p_{lc})$. In order to avoid division by 0, we substituted 1×10^{-100} to 0 for p_{ki}.

4.1.2 Results

Due to space limitation we show the results with $n(K) = 10$. Figure 1 shows 10 prototypes where the horizontal axis and the vertical axis represent the ID of a medical test and the probability that a value of a medical test becomes abnormal in % respectively. Medical tests that are judged important in the corresponding prototype are noted. The result of the clustering is {1,2,3},{4,5,6},7,8,9,10 when prototypes k and l are considered to belong to the same cluster if $\beta(k, l) \leq 10$.

From the figure, we see that prototype 7 and 9 largely differ from the rest. In prototype 7, medical tests which concern APO[3] show high probabilities, while in prototype 9, two medical tests show relatively much higher probabilities than others.

Domain experts commented that prototype 7 well-represents abnormal behavior on fat protein. However, most of the instances which belong to prototype 9 with high probability are measured at the second time in a day. In the data, medical tests which are measured in the second time in a day are almost fixed, thus we consider that prototype 9 models this tendency. Moreover, approximately half of all instances most probably belong to prototype 10, and 9578 instances among them have no abnormal test values, which

[3] APO is a kind of protein

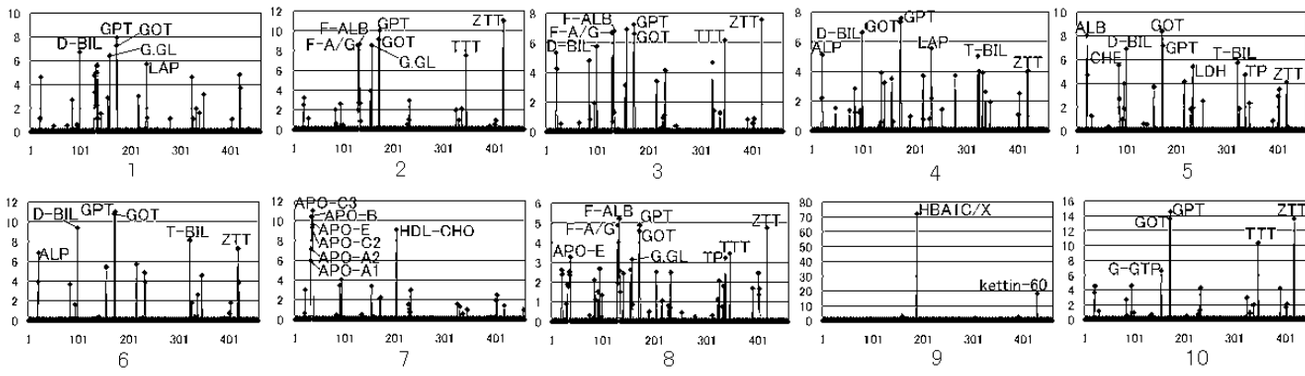


Figure 1. Generated Prototypes with the Single-Dice Model

can be hardly seen from the figure. Domain experts also commented that they would like to see more diversity in prototypes, since prototypes in the same cluster are similar. These results show that the single-dice model exhibits limited success.

4.2 Multi-Coin Model

4.2.1 Conditions

Domain experts inferred that the experiments in the previous section were not so successful since we employ all data and the single-dice model regards a normal value and a missing value equivalently. A virus marker test represents a medical test which reveals status of viruses and antibodies. In these paper, we call such a status a pattern. In these series of experiments, we followed the advice of domain experts and estimated a set of prototypes for each pattern of patients with B-type hepatitis. Table 1 shows the 13 patterns that we employed.

The total number of patients in these 13 patterns is 263 and the number of instances is 492. The drastic decrease of these in numbers is largely due to our restriction of patients to those who underwent four types of virus marker tests in a time of a day. We have also restricted the medical tests to eleven, and the number $n(K)$ of prototypes is 3.

4.2.2 Results

Figure 2 shows experimental results, where we omitted the results for patterns 2, 12, and 13 since each of them has a small number of instances. It should be also noted that pattern 12 and 13 cannot happen in the medical context. For each pattern, at most three prototypes are represented with different types of shading. The horizontal axis and the vertical axis represent 11 medical tests and the probability that a value of the corresponding medical test becomes abnormal. An occurrence probability of a prototype is shown above each graph.

Table 1. Patterns which show progress of B-type hepatitis. # represents the number of instances of the pattern

pattern	intuitive explanation	#
1	not infected	135
2	initial infection	2
3	virus active	63
4	antibody counter-activity	18
5	virus active and counter-existence	10
6	counter-activity and existence	5
7	virus not active and counter-existence	36
8	virus disappeared	59
9	virus not active	93
10	originally have counter-activity	24
11	originally have counter-existence	45
12	can't happen 1	1
13	can't happen 2	1

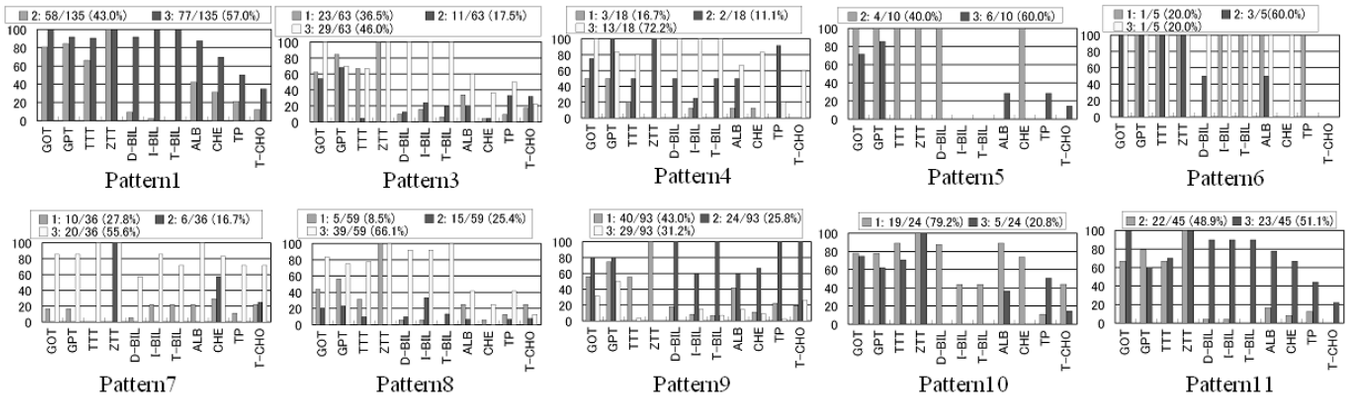


Figure 2. Generated Prototypes with the Multi-Coin Model

From the figure, we see that the probability that a value of the corresponding medical test becomes abnormal is typically much higher than those in figure 1. We consider that this is due to discrimination of a missing value from others, and modeling each medical test by a coin. The figure shows that the prototypes in a pattern typically exhibit diversity, and we mainly attribute this to the fact that they are extracted from patients of the same pattern.

Domain experts commented that many prototypes are valid, but mutually highly similar groups still exist. We have concluded that classification of test values to normal and abnormal is coarse, and a finer discretization is required.

4.3 Multi-Dice Model

4.3.1 Modification of Data

In the original data, many values of the virus maker tests are missing or doubtful[4]. Domain experts explained us that criteria of judgment for these tests have changed many times, and it is common to obtain impossible patterns and false judgments.

We again followed the advice of experts, and changed criteria for selecting data as well as completed missing values of the virus marker tests according to their domain rules. As the result, the new data set consists of 102 patients and 9,190 instances. Table 2 shows patterns after the data modification.

4.3.2 Results

As described in section 3.2, each medical test value can be labeled by discretization. We defined a set of labels[5] for each medical test.

Table 2. Patterns which show progress of B-type hepatitis after data modification

pattern	intuitive explanation	#
1	not infected	74
2	initial infection	113
3	virus active	3867
4	antibody counter-activity	1157
5	virus not active	3419
6	counter-existence	254
7	virus disappeared	368
8	originally have counter-existence	10
9	originally have counter-activity	22

Figure 3 shows the obtained prototypes. In each pattern, three prototypes are shown at the leftmost part, the middle part, and the rightmost part. The horizontal axis and the vertical axis represent 11 medical tests and the probability that a value of the corresponding medical test becomes one of the discretized labels. Note that we employ a ratio-bar graph for each medical test of a prototype since the add-sum of the probabilities of labels is 100 %. An occurrence probability of a prototype is shown below each graph.

We see more diversity in prototypes for each pattern than in figure 2, and we attribute this to the more detailed discretization than in the multi-coin model as well as the additional pre-processing of data. Domain experts commented that many of the obtained prototypes are highly-readable and have clear meaning in the medical context. Also they are eager to see application of this method to prediction problems such as effectiveness of interferon[6] and progress

[4]These are mainly due to uncertain judgment of doubtful positive results and doubtful negative results

[5]Labels are represented by vL (very Low), L (Low), N (Normal), H (High), vH (very High), and uH (ultra High). A label which is nearer to Normal shows that the status of a patient is less severe. A set of labels depends on a medical test: {VL,L,N,H,VH} for {ALB, TP, T-CHO, CHE}, and {N, H, VH, UH} for {GOT, GPT, ZTT, TTT, T-BIL, D-BIL, I-BIL}

[6]Roughly speaking, interferon is a drug that kills virus

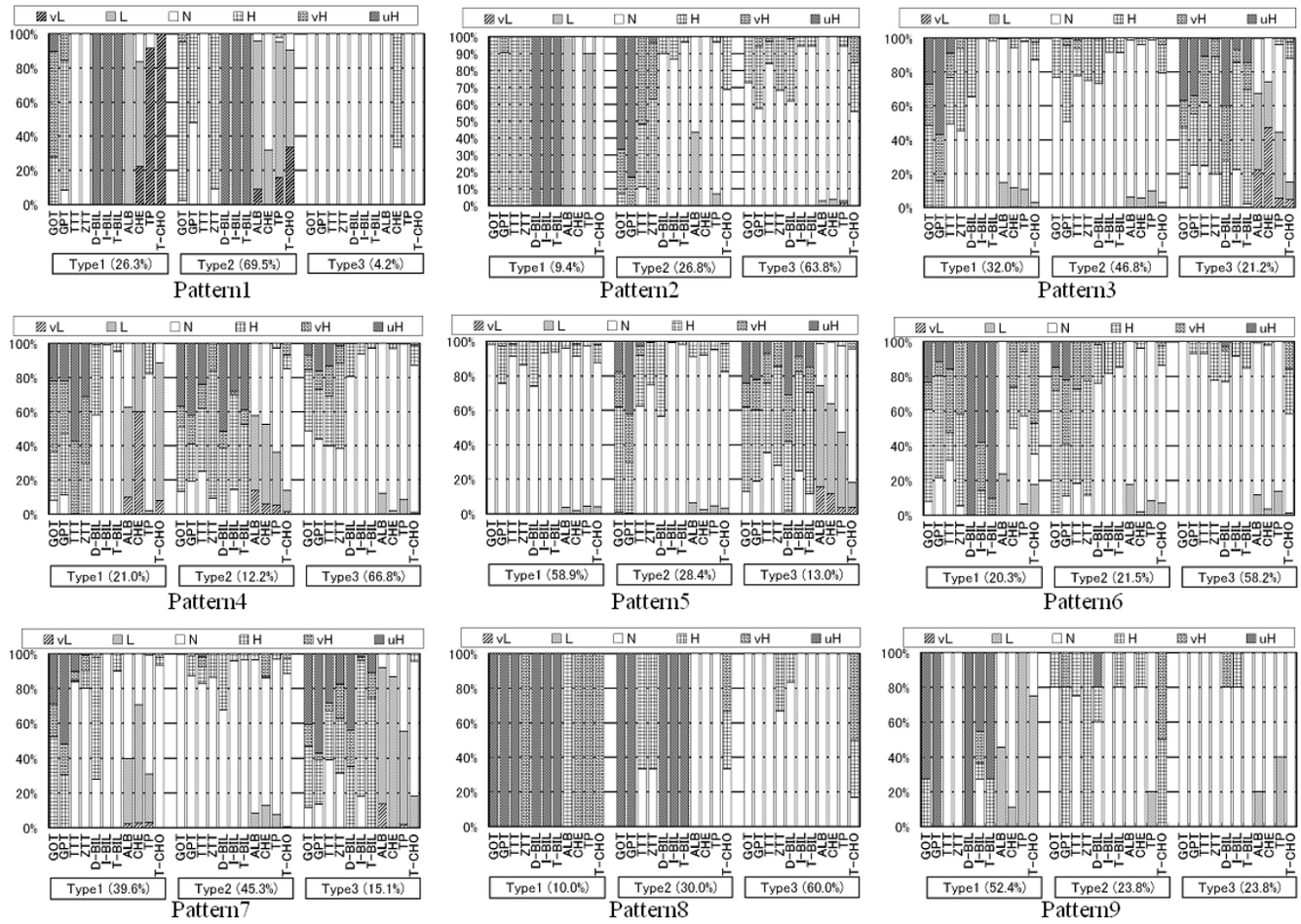


Figure 3. Generated Prototypes with the Multi-Dice Model

of hepatitis disease. According to them, pattern 2, 3, 5, and 7 show valid prototypes which are clearly separated. Prototypes in pattern 1 seemed inappropriate at first, but it was due to the fact that the corresponding data set consists of a small number of peculiar patients. Our multi-dice model is also effective in finding such peculiar patients.

4.3.3 Application to Visualization

Figure 4 shows pairs of a pattern and a prototype k_j with the conditional probability $p(k_j|\mathrm{x}_i)$ of a patient x_i in chronological order. From the figure, we see that patient 446 belongs to the pattern active and to prototype 2 by 100 % on April 20, 1987. Since the prototype, which can be found on the upper-right of figure 3, shows that most test values are normal, the patient showed relatively normal results though s/he belonged to the pattern active on that day. However, on October 24, 1990, s/he became a little worse since s/he belongs to the pattern counter-activity and the prototype 3. From subsequent pairs of (pattern, prototype), we see that the patient belongs to (active, worse), (not active, a little bad), (not active, very good), and (virus disappeared, a little worse).

ID: 446	87_ 4/20	·····	90_10/24	·····	91_ 1/16
	3-2(100.0)		4-3(100.0)		3-1(99.1)
Date	3-1(0.0)				3-2(0.9)
	3-3(0.0)	Pattern-Type(Probability)			
Patient	93_ 5/12	·····	93_ 8/ 4	·····	96_ 3/13
	5-2(98.8)		5-1(100.0)		7-2(100.0)
	5-3(0.7)		5-2(0.0)		

Figure 4. Visualization of a Patient's History

Our method based on the multi-dice model transforms practically unreadable test values to highly readable prototypes, and is thus effective in recognizing chronological progress of a disease of a patient. We admit that some prototypes are still a little counter-intuitive to domain experts, but they consider that we should re-classify patterns using

further domain knowledge as well as omit peculiar patients as described in section 4.3.2. Our active mining of chronic hepatitis data continues, and we believe that our method is an effective tool in this endeavor.

5 Conclusions

We have derived prototypes of medical test values based on estimation of probabilistic mixture models. The process consists of a series of trials and errors with the help of domain experts. The result was successful due to careful selection and modification of data as well as our multi-dice model as a representation of a prototype. For the domain experts, most of the obtained prototypes are highly-readable and have clear meaning in the medical context.

The experts are eager to see further progress in prediction problems. Since current prototypes are informative and not discriminative [6], they would be effective if they can describe the target concept. We consider that the key to success for a wider range of problems will be incorporation of class information as well as further pre-processing of data.

Acknowledgement

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 et al. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI/MIT Press, Menlo Park., Calif., 1996.

[2] P. Berka. ECML/PKDD 2002 discovery challenge, download data about hepatitis. http://lisp.vse.cz/ challenge/ecmlpkdd2002/, 2002. (current September 28th, 2002).

[3] I. V. Cadez, P. Smyth, and H. Mannila. Probabilistic modeling of transaction data with applications to profiling, visualization, and prediction. In *Proc. Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 37–46, 2001.

[4] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–38, January 1977.

[5] W. Feller. *An Introduction to Probability Theory and Its Applications, Volume 1*. John Wiley & Sons, 1957.

[6] Y. D. Rubinstein and T. Hastie. Discriminative vs informative learning. In *Proc. Third International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 49–53, 1997.

[7] S. Russell and P. Norvig. *Artificial Intelligence, A Modern Approach*. Prentice Hall, Upper Saddle River, N. J., 1995.

Development of Generic Search Method Based on Transformation Invariance

Fuminori Adachi, Takashi Washio, Hiroshi Motoda and *Hidemitsu Hanafusa
I.S.I.R., Osaka University, {adachi, washio, motoda}@ar.sanken.osaka-u.ac.jp
*INSS Inc., hanafusa@inss.co.jp

Abstract

The needs of efficient and flexible information retrieval on multi-structural data stored in database and network are significantly growing. Especially, its flexibility plays one of key roles to acquire relevant information under interactions among domain experts, data mining experts and data mining tools in active mining process. However, most of the existing approaches are dedicated to each content and data structure respectively, e.g., relational database and natural text. In this work, we propose a generic information retrieval method directly applicable to various types of contents and data structures. The power of this approach comes from the use of the generic and invariant feature information obtained from byte patterns in the files through some mathematical transformation. The experimental evaluation of the proposed approach for both artificial and real data indicates its high feasibility.

1 Introduction

The recent progress of information technology increases the variety of the data structure in addition to their amount accumulated in the database and the network. The flexible information retrieval on multi-structured data stored in the computers is crucial to acquire relevant information under interactions among domain experts, data mining experts and data mining tools. However, the state of the art remains within the retrieval for each specific data structure, e.g., natural text, relational data and sequential data [1], [2], [3]. Accordingly, the retrieval on mixed structured data such as multimedia data containing documents, pictures and sounds requires the combined use of the retrieval mechanisms where each is dedicated to a data type respectively [4], [5]. Because of this nature, the current approach increases the cost and the work of the development and the maintenance of the retrieval system.

To alleviate this difficulty, we propose a novel retrieval approach to use the most basic nature of the data representation. Any data is represented by the sequence of bits or bytes. Accordingly, a generic retrieval method is established if a set of data which is mutually similar on this basic representation can be appropriately searched. The main issue on the development is the definition of the similarity in the low level representation which appropriately corresponds to the similarity on the content level. Though the perfect correspondence may be hardly obtained, the following points are considered to enhance the feasibility of our proposal.

(1) Commonly seen byte sequences in approximately similar order and length are searched.
(2) The judgment of the similarity is not significantly affected by the location of the patterns in the byte sequences.
(3) The judgment of the similarity is not significantly affected by the noise and the slight difference in the byte sequences.
(4) The mutual similarity of the entire files is evaluated by the frequency of the similar byte sequences shared among the files.
(5) The similar byte sequences shared by most of the files are removed to evaluate the similarity among the files as they do not characterize the specific similarity.

In this work, an approach to the generic method to retrieve similar files in terms of the byte sequences is studied. A certain mathematical transform on the byte sequences is used by treating each byte as a numeral. This can extract invariant characters of the sequences and the files to meet the aforementioned consideration. The basic performance of the proposed approach is evaluated through numerical experiments and a realistic application to the retrieval of raw binary format data of a word processor.

2 Principle of Similarity Judgment

The aforementioned point (1) is easily achieved by the direct comparison among byte sequences. However, the point (2) requires a type of comparison among sequences that is invariant against the shit of the sequences. If the direct pair wise comparison between all subsequences selected from two sequences respectively is applied, the computational time is $O(n_1^2 n_2^2)$ where n_1 and n_2 are the numbers of bytes in the two sequences. To avoid this high complexity in

practical sense, our approach applies a mathematical transform to the byte sequence in each file. The transform has the property of "shift invariance" where the value obtained through the transform is hardly changed against the shift of the sequence. To address the point (3), the result of the transform should be quite robust against the noise and slight difference in the sequence. Moreover, the transform must be conducted within practically tractable time. One of the representative mathematical transform to suffice these requirements is the Fast Fourier Transform (FFT) [8]. It requires only computation time of O(n log n) in theory when the length of the byte sequence is n, and number of practical methods for implementation are available. In addition, the resultant coefficients can be compressed into the amount of 50% of the original if only their absolute values are retained. However, when the transform is applied to very long sequences or sub-sequences contained in a large file where each part of the file indicates a specific meaning, the local characters of the byte sequence reflecting the meaning in the contents level will overlap with the local characters of the other part. Accordingly, we partition the byte sequence in a file into an appropriate length, and apply the FFT to each part to derive a feature vector consisting the absolute values of the Fourier coefficients.

Because this approach is quite novel, the basic feasibility and the characteristics of the proposed method have been checked though some numerical experiments on some pieces of byte sequences in advance. In the experiment, the length of each byte sequence is chosen to be 8 bytes because it is the length of byte sequences to represent a word in various languages in standard. Though each byte takes a value in the range of [0, 255], a number 128 is subtracted from the value to eliminate the bias of the FTT coefficient of order 0. First, we shift the byte sequences to the left randomly, and the bytes out of the edge are located in the right in the same order. Thus, the byte sequences are shifted in circular manners. Because of the mathematical nature of FFT, i.e., shift invariance, we observed that this did not cause any change of the transformed coefficients. Next, the effect of the random replacement of some bytes are evaluated. Table 1 exemplifies the effects of the replacement in a basic sequence "26dy10mo" on the transformed coefficients. The distance in the table represent the Hamming distance, i.e., the number of the different bytes from the original. The coefficients from f_5 to f_8 are omitted due to the symmetry. In general, only n/2+1 coefficients for an even number n and (n+1)/2 for an odd number n are retained. The numbers of the coefficients are quite similar within the Hamming distance 2 in many cases. However, they can be different to some extent even in the case of distance 2 such as "(LF)5dy10mo" where the value of "(LF)" is quite different from that of "2". Accordingly, some counter measure to absorb this type of change or noise in the similarity judgment must be introduced.

Table 1 Effect of byte replacements on FFT coefs.

Sequences	f0	f1	f2	f3	f4	Distance
26dy10mo	144	112.9	345.6	103.8	108	0
20dy10mo	150	112.4	350.7	103.9	102	1
19dy10mo	142	113.8	343.6	103.1	112	2
(LF)5dy10mo	174	89.9	361.2	136.2	156	2
(LF)5dy11mo	178	86.6	364.4	137.3	152	3
(LF)5dy09mo	180	88.6	365.8	136.8	152	4

The method taken to enhance the robustness against the replacements in this work is the discretization of the FFT coefficients. If the coefficients are discretized in an appropriate manner, the slight differences of the coefficient vales do not affect the similarity judgment of the byte sequence. An important issue is the criterion to define the threshold values for discretization. The most efficient way to define the thresholds is that the coefficient obtained from an arbitrary sequence falls into an interval under an identical probability. To define the thresholds of the coefficient in every order for a certain length of byte sequences, i.e., the length n, we calculated coefficient value distribution for all 2^{8n} byte sequences. This computation is not tractable, however in practice, this is quite easily achieved by using the symmetric characteristics of FFT coefficients on various sequence patterns. Upon the obtained coefficient distribution for every order, (m-1) threshold values are defined where every interval covers the identical probability 1/m in the appearance of a coefficient of every order. When the number of m is small, the character of each byte sequence does not become significant due to the rough discretization. We tested various number m, and chose the value m=16 empirically which is sufficient to characterize the similarity of the byte sequence in generic means. Through this process, the information of a FFT coefficient for every order is compressed into 16 labels. In summary, a feature vector consisting of n/2+1 or (n+1)/2 elements for an even or odd number n is derived where each element is one of the 16 labels.

Moreover, the moving window of a fixed length byte sequence is applied to generate a set of feature vectors for a file. First, a feature vector of the byte sequence of a length n at the beginning of the file is calculated. Then another feature vector of the sequence having the same length n but shifted with one byte toward the end of the file is calculated. This procedure is repeated until the feature vector of the last sequence at the end of the file is obtained. This approach also enhance the

robustness of the similarity judgment among files. For example, the feature vectors of the first 8 bytes windows of "26dy10mo02yr" and "(LF)5dy10mo02 yr" are quite different as shown in Table 1. However, the feature vectors for the 8 bytes windows shifted by one byte, i.e., "6dy10mn0" and "5dy10mn0", are mutually very similar. Furthermore, the vectors for the windows shifted by two bytes become identical because both byte sequences are "dy10mo02". This moving window approach enables the frequency counting of the parts having similar patterns among files. Thus, the point (4) mentioned in the first section is addressed where the mutual similarity of the entire files is evaluated by the frequency of the similar byte sequences shared among the files. To address the point (5), the feature vectors which is obtained from a given set of files more than a certain frequency threshold are registered as ineffectual vectors, and such ineffectual vectors are not used in the stage of the file retrieval.

3. Fast Algorithm of Retrieval

The data structure to store the feature vectors for given vast number of files must be well organized to perform the efficient file retrieval based on the similarity of the byte sequences. The approach taken in this work is the inversed file indexing method which is popular and known to be the most efficient in terms of retrieval time [3]. Through the procedure described in the former section, the correspondence from each file to a set of feature vectors derived from the file is obtained. Based on this information, the inversed data indexing from each feature vector to a set of files, that produced the vector, is derived. The data containing this inversed indexing information is called "inversed indexing data". By using the inversed correspondence in this data, all files containing patterns which are similar with a given feature vector are enumerated efficiently.

Figure 1 outlines our retrieval system. The path represented by solid arrows is the aforementioned preprocessing. The "Data Extraction" part applies the moving window extraction of byte sequences to each files in a given set of data files. The extracted byte sequences are transformed by FFT in the "Mathematical Transformation" part. The "Vector Discretization" part discretizes the resulted coefficients by the given thresholds, and the feature vectors are generated. The "Vector Summarization" part produces the correspondence data from each file to feature vectors while removing the redundant feature vectors among the vectors derived from each file. Finally, the "Inversed Indexing" part derives the inverse correspondence data from each feature vector to files together with the "ineffectual vectors list".

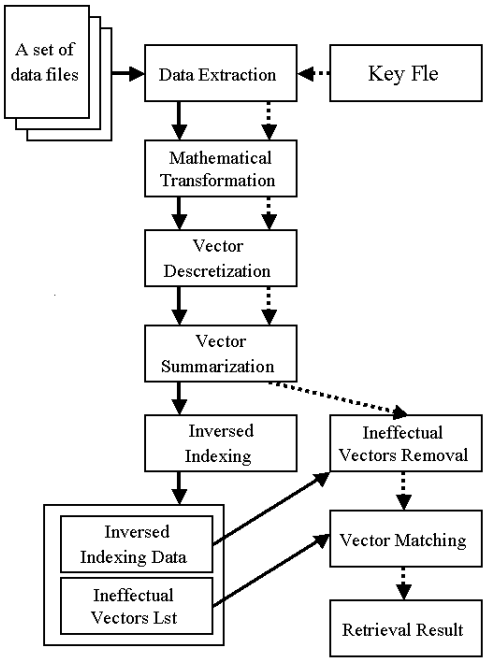


Figure 1 Outline of retrieval system

The file retrieval is conducted along the path represented by the dashed arrows. A retrieval key file having the content which is objective or similar to the objective is given to the "Data Extraction" part, and the identical information processing with the former paragraph derives the set of the feature vectors of the key file. Subsequently, the ineffectual vectors are removed from the set in the "Ineffectual Vectors Removal" part. Finally, the files corresponding to the feature vectors in the set are enumerated based on the inverse correspondence data in the "Vector Matching" part. To focus the retrieval result to only files having strong relevance with the key files, a frequency threshold value is applied in the enumeration. If the frequency of the vector matching is less than the threshold for a file, the file is not retained in the retrieval result. Moreover, the result is sorted in the order of the matching frequency.

4. Basic Performance Evaluation

A program based on the proposed method has been developed, and its basic performance was evaluated by using artificial data sets. The specification of the computer used in this experiment is CPU: AMD Athlon 1400MHz, RAM: PC2100 DDRSDRAM 348MB, HDD: Seagate ST340824A and OS: LASER5 Linux 7.1. 500 files having the normal distribution in their sizes were generated. Their average was 30KB and the standard deviation 10KB. The byte data in each file were generated by using the uniform random distribution. Next, 5 specific sequences in the length of

16 bytes, which were labeled as No.1, …, 5, were embedded in each file. They were embedded not to mutually overlap, and moreover the nonexistence of the sequences identical with these 5 sequences is confirmed. The parameters of the generation of the feature vectors are the moving window size of 8 bytes, the 16 level of discretization of the FFT coefficients for each order and 70% for the threshold frequency to determine the ineffectual vector.

Table 2 Retrieval by key file No.1

Sequence No.	Thre-shold	Retrieved Files	Correct Files	Pre-cision	Recall	Comp Time
1	1.0	250	250	1.00	1.00	1.6
	0.25	261	250	0.96	1.00	
	0.125	344	250	0.73	1.00	

Table 3 Retrieval by shifted key file No.1

Sequence No.	Thre-shold	Retrieved Files	Correct Files	Pre-cision	Recall	Comp. Time
1	0.66	2	2	1.00	0.01	0.7
	0.55	37	37	1.00	0.15	
	0.44	250	250	1.00	1.00	
	0.33	252	250	0.99	1.00	
	0.22	266	250	0.94	1.00	
	0.11	326	250	0.77	1.00	

The performance indices used in the experiment is the precision and the recall. In ideal situation, both values are close to 1. However, they have a trade off relation in general. Table 2 shows the performance of the retrieval by the key file consisting of the sequence No.1. The thresholds in the table are the frequency levels of the feature vector matching to evaluate the similarity of the files in the "Vector Matching" part in Fig.1. The high value of the threshold retains only highly similar files. The sequence No.1 is embedded in the 250 files among 500 test files. This is reflected in the result of the threshold equal to 1.0, i.e., the key file consisting of the sequence No.1 is certainly included in these files as a subsequence. In the lower value of the threshold, some files containing similar subsequence with the sequence No.1 are also retrieved. Thus, the precision decreases. In this regard, our proposing approach has a characteristic to retrieve a specified key pattern similarly to the conventional keyword retrieval when the threshold is high.

Table 3 shows the result of the retrieval where the key sequence No.1 is shifted randomly in circular manner. Because the length of the embedded sequences and the key sequence is 16 bytes, but that of the moving window for FFT is only 8 bytes, the FFT coefficients do not remain identical even under its shift invariance characteristics. Accordingly, the feature vector of the key sequence does not match with these of the embedded sequences. However, the coefficients of FFT reflects their partial similarity to some extent, and thus the excellent combination of the values of the precision and the recall is obtained under the threshold values around 0.2 - 0.4. Similar results were obtained in case of the other key sequences. In contrast, when we applied the conventional retrieval approach based on the direct matching, very low values of the precision and the recall were obtained for every threshold levels. Table 4 represents the results for noisy data. 2 bytes randomly chosen in each original 16 bytes sequence are replaced by random numbers. Similarly to the former experiment, the excellent combination of the precision and the recall was obtained for every key sequence under the threshold value of 0.3 – 0.5. If the distortion on the embedded sequences by the replacement becomes larger, i.e., the increase of the number of bytes to replace, the values of precision and the recall decreases. But, the sufficient robustness of the proposed retrieval approach under the random replacement of 3 or 4 bytes in the 16 bytes sequence has been confirmed through the experiments.

Table4 Retrieval on Noisy Data

Sequence No.	Thre-shold	Retrieved Files	Correct Files	Pre-cision	Recall	Comp Time
1	0.33	3	2	0.67	0.01	0.9
	0.22	27	18	0.67	0.07	
	0.11	159	92	0.59	0.37	
2	0.77	1	1	1.00	0.01	1.2
	0.66	15	15	1.00	0.04	
	0.55	125	125	1.00	1.00	
	0.22	140	125	0.89	1.00	
	0.11	203	125	0.62	1.00	
3	0.625	1	1	1.00	0.01	1.0
	0.500	3	3	1.00	0.03	
	0.375	31	28	0.90	0.28	
	0.250	120	100	0.83	1.00	
	0.125	229	100	0.44	1.00	
4	0.375	1	1	1.00	0.02	1.1
	0.250	38	14	0.37	0.28	
	0.125	178	50	0.28	1.00	
5	0.66	3	3	1.00	0.12	1.2
	0.55	25	25	1.00	1.00	
	0.22	34	25	0.74	1.00	
	0.11	127	25	0.20	1.00	

The computation time to finish a retrieval for a given key file was less than 1 second due to the inverse indexing approach. Thus, the proposed method is highly practical for considerably large scale application. In short summary, the basic function of our approach subsumes the function of the conventional retrieval approach, because the conventional retrieval is performed by setting the frequency threshold of the

feature vector matching at a high value. Moreover, this approach can retrieve the files having some generic similarity.

Table 5 Retrieval on semi-real world data

Key File No.100	Key File No.500	Key File No.1000	Key File No.1500	Key File No.2000
100	500	1000	1500	2000
102	676	789	1499	2001
99	664	979	1494	1999
104	508	648	1498	1995
96	554	999	1502	2158
97	503	967	1497	2258
105	579	997	1496	1868
106	561	856	1503	2019
103	543	852	1504	1989
98	485	543	1506	1877
109	471	513	1508	2208
113	541	998	1501	2171
107	642	857	1495	2008
92	636	855	1507	2020
116	513	695	1505	2004
101	553	676	1511	1855
93	789	672	1513	2154
108	498	591	1510	2135
95	567	508	1509	2079
111	525	1084	1489	2068
110	504	1006	1492	2207
112	486	837	1491	2061
117	852	982	1515	2022
88	848	975	1493	2269
114	678	909	1488	2192
120	604	896	1512	2126
94	477	534	1519	1990
128	933	858	1486	1958
118	870	627	1483	1908
115	689	500	1518	2150
79	571	995	1485	2017
127	856	991	1514	1957
125	836	948	1487	1858
121	695	563	1528	1857
81	537	541	1490	2257
89	897	984	1521	2181
90	562	965	1516	2160
122	558	615	1482	2009
138	524	996	1517	2005
87	869	986	1524	1961
141	615	854	1525	2245
139	544	992	1484	2153
78	817	974	1520	2121
142	555	814	1527	2007
124	497	773	1526	1986
126	959	760	1480	2157
85	873	677	1522	2096
140	679	562	1476	2089
83	677	994	1523	2081
123	712	993	1481	1994
Ave. 108.22	Ave. 641.64	Ave. 823.60	Ave. 1503.44	Ave. 2058.80
t= -148.6	t= -70.8	t= - 44.2	t= 55.0	t= 136.1
0.642 sec	0.466 sec	0.422 sec	0.844 sec	0.370 sec

5. Evaluation on Word Processor Files

The practical performance of our proposed method is evaluated by using semi-real world data. The data is a set of 2253 word processor files having Microsoft Word doc format. Their average size is around 20KB, and each contains around 600 characters in form of a document. To evaluate the ability to retrieve similar content files within our proposing approach, the raw content data are converted to have some similar relations among some files. Initially, a seed file is selected from the original set of word processor files and numbered as No.1. Then, another file X is randomly chosen from the raw file set, and a sequence consisting 16 characters in the file is selected from the file. Then, a randomly chosen part consisting of 16 character sequence in the original file No.1 is overwritten by the sequence selected from the file X, and the new file is numbered as No.2. Starting from this stage, a part of 16 characters randomly chosen in the file No. n is overwritten by the sequence of 16 characters selected from a randomly chosen file X, and the new file is numbered as No. n+1. This process is repeated 2253 times to gradually and randomly change the original seed file and newly generate similar files. As a consequence, 2253 files in total are generated where the files having close number have some similarity.

Based on this semi-real world data, the inversed indexing data and ineffectual vector list are generated in the preprocessing stage of our approach. Subsequently, 5 key files arbitrary chosen from the semi-real world files are used to retrieve their similar files. Each key file is given to the retrieval system and processed along the dashed line in Fig.1. Table 5 show the result of the top 50 retrieved files in the order of the similarity in terms of the feature vector matching. The result clearly shows that the files having close number to the key file are retrieved. Some files are missing to be retrieved even when their numbers are closer to the number of the given key file. This is because the character sequence for the replacement can be quite different from the original overwritten sequence in terms of numerical series data, and this replacement significantly affects the coefficients of FFT to from the feature vectors. This effect has been already shown in the example of the feature vectors of “26dy10mo02yr” and “(LF)5dy10mo02yr” in Table 1. Though the moving window approach alleviates this type of distortion on the judgment of similarity, the judgment is infected to some extent even under this approach. The third row from the bottom in the table indicates the average of the top 50 files’ numbers, and the second row from the bottom shows the t-value on the deviation of the average of the top 50 files’ numbers from the expected average of the uniformly sampled 50 files’

numbers, i.e., 1126.5. According to the Central Limit Theorem, the average approximately follows the normal distribution N(1126.5, 105750) under the uniform sampling. The absolute t-value more than 3.3 indicates that the probability that the files retrieved follow the uniform distribution is less than 0.001. Therefore, the distributions of the retrieval results are sufficiently skewed around the key files in the sense of the similarity. The bottom row represents the computation time to retrieve the 50 files for each key files. The 50 similar files are retrieved within a second among the 2253 doc files for each key file. The difference of the time for retrieval is due to the difference of the number of the feature vectors which is not ineffectual for each key file. For example, the number of the effective feature vector of the key fille No.1500 is 1474 while it is only 593 for the key file No. 2000. This difference is reflected to the retrieval time. The retrieval time is almost linear with the number of the effective feature vectors of each key file.

6. Discussion and Related Work

The signature files method to use moving windows of byte sequences having a fixed length in the files has been proposed for file retrieval [2]. This method compresses each byte sequence in incomplete and irreversible fashion by introducing hash functions, and efficiently focuses on similar key sequence patterns on the reduced size of binary signature data. However, the direct matching of key sequence is required at the final stage of the retrieval to achieve the complete retrieval because of the incompleteness of the signature matching. On the other hand, the inversed indexing approach where the files containing each key are listed in advance are often used for the practically fast retrieval [3]. One of the representative system is Namazu for Japanese documents [9]. Though this approach needs considerably large space for the indexing data storage, the recent increase of the capacity of the storage devices is alleviating this difficulty. However, this approach is for the complete matching on the files such as text documents.

In contrast, our proposing method applies a mathematical transform having some invariance and compression properties to retain the information of certain similarities among files rather than the ordinary hash compression function. Because of the nature of the mathematical transform, the complete matching is easily achieved in our framework if the threshold value for feature vector matching is taken at a high frequency. Moreover, the incomplete matching to retrieve files containing similar patterns in terms of the invariance and robustness of the transform is also achieved by applying the lower threshold value. The efficiency of the retrieval is comparable with the ordinary inversed indexing approach because our approach also uses the inversed indexing on the representation of feature vectors.

7. Conclusion

In this work, a generic retrieval approach for the data, where one dimensional byte sequences reflect the contents of the data, is proposed. The examples of the data are the ordinary text files, word processor files and sound data files. The proposed approach covers the most advantage of the conventional approaches. The next issue is to extend this approach to multi-dimensional data such as image data and 3D data where the information of the contents are not reflected in the byte sequences in straight forward manner.

Reference

[1] Baeza-Yates, R.A.: String Searching Algorithms, Information Retrieval, Data Structures & Algorithms, Chapter 10, ed. Baeza-Yates, R.A., New Jersey: Prentice Hall, pp. 219-240 (1992).

[2] Faloutsos, C: Signature Files, Information Retrieval, Data Structures & Algorithms, Chapter 4, ed. Baeza-Yates, R.A., New Jersey: Prentice Hall, pp. 44-65 (1992).

[3] Harman, D., Fox, E. and Baeza-Yates, R.A.: Inverted Files, Information Retrieval, Data Structures & Algorithms, Chapter 3, ed. Baeza-Yates, R.A., New Jersey: Prentice Hall, pp. 28-43 (1992).

[4] Ogle, V.E., Stonebraker, M: Chabot: Retrieval from a Relational Database of Images, IEEE Computer, Vol. 28, No. 9, pp.1-18 (1995).

[5] Faloutsos, C., Equitz, W., Flickner, M., Niblack, W., Petkovic, D., Barber, R.: Efficient and Effective Querying by Image Content, Journal of Intelligence Information Systems, 3, 3/4, pp.231-262 (1994).

[6] Fox, C: Lexical Analysis and Stoplists, Information Retrieval, Data Structures & Algorithms, Chapter 7, ed. Baeza-Yates, R.A., New Jersey: Prentice Hall, pp. 102-130 (1992).

[7] Salton, G. and McGill, M.J.: Introduction to Modern Information Retrieval, McGraw-Hill Book Company (1983).

[8] Digital Signal Processing, The Institute of Electronics, Information and Communication Engineers (IEICE) 10th Ed., Gihoudou, pp.49-61 (1983) (in Japanese).

[9] http://www.namazu.org/

Information Extraction for On-line Job Advertisements

Kwok-Chung Au and Kwok-Wai Cheung
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
{william, henryau}@comp.hkbu.edu.hk

Abstract

The Web has widely been used as a low-cost and yet effective way to disseminate various kind of information, where job advertisements (ads) posting is one of the typical examples. Unlike many typical Web documents, on-line job ads contain grammatical, telegraphic as well as ungrammatical text. Their writing styles and layout structures are much less formal which make the traditional NLP-based information extraction techiques simply fail. The pattern matching approach is commonly used instead for this kind of documents. In this paper, an information extraction system using the pattern matching approach is described. The system represents the extraction pattern using lexical and typesetting information and the extraction rules are derived automatically via an induction process. Via experiments, we show that the proposed system can automatically choose the correct rule representation via the induction. In particular, the system can successfully extract company names from on-line job ads with an extraction accuracy of 92.13%.

1 Introduction

The Web has widely been used as a low-cost and yet effective way to disseminate various kind of information. However, the rapid growth of the Web leads to the well-known *information overload* problem and thus searching relevant information embedded in heterogeneous on-line repositories becomes a non-trival process. To alleviate the problem, some sophisticated tools (information extraction systems) for extracting semantic information from Web pages are inevitably required.

Information extraction (IE), originated from the natural language processing community, has a history of more than a decade and many related systems have been built [8] with the objective to extract target information from a large collection of documents. Related IE systems proposed in the literature include Rapier [2], WHISK [10] and SRV [4]. In terms of *representation*, these systems use a set of extraction rules to model the local patterns of the target information for identification. For extracting information from grammatical free text (e.g., articles and news), part-of-speech (POS) tagging (using a syntactic analyzer) and semantic tagging (using some dictionaries) are commonly adopted to first preprocess the document. Then, extraction rules can be specified using syntactic and semantic constraints for the local patterns.

For text with HTML layout tags, the use of HTML tags are found to be effective. Related extraction systems are commonly called *wrappers*. WIEN [6] is a wrapper which consists of various grammar classes of HTML tags, each corresponds to the extraction of information tagged in a particular grammatical structure. SoftMealy [5] summarizes the grammatical structures using a finite-state transducers. STALKER [9] allows extraction rules to be integrated in a hierarchical manner where the extraction rules can be embedded into one another.

For on-line advertisements like job ads, their contents seldom contain complete grammatical sentences. Instead, short and concise phrases with some HTML tags revealing the documents' layout (semi-structured text) are found. Most of the aforementioned techniques which use POS tagging will simply fail. On the other hand, performing the extraction solely based on the HTML tags is also insufficient. For example, it is common for a list of job requirements to be included within a paragraph marked with <p>. It is by no means the HTML tags can help extracting the individual information items out. The system we described in this paper is developed mainly for extracting information from these kind of advertisement documents.

The pattern matching approach is used in our system to be described in this paper where we represent the extraction patterns using only the words and their typesetting attributes. Rapier [2] is one of the extraction systems closest to our proposed system and has also been applied to job ads. The main difference between our system and Rapier is that Rapier uses POS information while we use typesetting

attributes instead. It turns out that quite a number of specially designed pattern classes (later on called token classes in Section 2.1) are required and they are unique to the application. Due to the different representations, the underlying rule induction algorithm is also different.

We have evaluated our system using a dataset containing job ads (∼500) collected from a local on-line classified post and obtained an accuracy of 92.13%. The remaining of the paper is organized as follow. In Section 2, we describe our rule representation and then the rule induction algorithm in Section 3. Section 4 is devoted to the experiment setup and evaluation. Section 5 concludes the paper.

2 Representation

Our proposed system adopts the pattern matching approach for information extraction. So, one of the important issue is the choice of rule representation, which greatly affect the power of the system. We model patterns of target information using a composition of:

Tokens which are derived from a set of predefined *token classes*, each characterizing a particular type of document items, and

Word Lists which are formed by collections of possible words.

2.1 Token Classes

Our system uses eleven different token classes inspired from Hsu *et al.*'s work [5]. Six of the classes are related to different types of typesetting and thus called *typesetting tokens*. [1] The remaining five are related to other item types and are here called *special tokens*.

- *Typesetting Tokens:*

  Uppercase Strings: `[Upper,#num]`, e.g., "HONG KONG".

  Lowercase Strings: `[Lower,#num]`, e.g., "computer science".

  Capitalized Strings: `[Cap,#num]`, e.g., "Baptist University".

  Mixed Strings: `[Mixed,#num]`, e.g., "MicroStation".

  sCapitalized Strings: `[sCap, #num]`, e.g., "i-Cable, 7-Eleven".

  sUpper Strings: `[sCap, #num]`, e.g., "i-CABLE".

- *Special Tokens:*

  Lexicon: e.g., `[<Company>]`. (to match strings predefined in the associated lexicon file)

  Punctuation Symbols: `[Punct,#num]`, e.g., ",", ";", ...

  Numeric Strings: `[Number,#num]`, e.g., "2000".

  Newline Character: `[NEWLINE]`. Match only the newline character

  HTML Tags: e.g., `["<HTML>"]`. (default case insensitive)

where `#num` denotes the occurrence parameter of a token and is an integer greater than zeros. It corresponds to the upper bound of consecutive occurrences of the associated token instance.

2.2 Word Lists

Using solely the token classes for rule representation will result in systems which are over-generalized to extract all the formatted entities inside a document and thus a high false alarm rate. To reduce the false alarm rate, we use word lists as another cue orthogonal to token classes to restrict the over-generalized token-based extraction rules. To differentiate from the Lexicon token class, these word lists contain some more subtle vocabularies specific to particular application domains (e.g., company names, job natures) while the Lexicon token class refers to some more obvious vocabularies which can be easily created in an *a priori* manner like sets of synomyns.

2.3 Rule Modeling

Based on the token classes and their associated word lists, *extraction rules* can be defined to characterize the target information to be extracted. In our system, an extraction rule consists of five components, namely

1. the prefix sub-pattern right ahead of the target pattern, **b**,
2. the beginning sub-pattern of the target pattern, **B**,
3. the ending sub-pattern of the target pattern, **E**,
4. the suffix sub-pattern right after the target pattern, **e**, and finally
5. the maximum number of strings allowed in between **B**and **E**, **g**.

See Figure 1 for an illustrated example.

[1] Note that there are some token classes which are specifi cally designed for modeling company names.

3 Rule Induction

Based on the rule representation, one can manually create a set of extraction rules using domain knowledge. Thus, domain experts with prior experience in designing the extraction rules will be required, and yet trial-and-error iterations are inevitable in most of the cases. Inductive learning techniques have long been known to be useful for automating the rule generation process. For our system, the set of extraction rules is induced using a bottom-up learning process based on manually tagged training data, as described in Section 3.1.

3.1 Rule Learning

In order to induce the extraction rules, a set of on-line documents with the target information being tagged is required. Based on the tagged documents, individual target items together with their neighboring items can readily be obtained and form the training dataset for the subsequent learning. The learning process we adopted involves three stages, namely, *tokenization*, *initial rule set generation* and *rule generalization*.

3.1.1 Tokenization

This is the first stage where words within each training sample are tokenized according to the token definition. Some tokens are exclusive to each other while some are not. So, there are cases where an item can be represented by more than one token classes. For example, the word "Company" can be tokenized as a Lexicon token or a Capitalized token. We resolve these conflicts using a pre-defined precedence order. The logic behind our precedence order is to have the more restricted token classes to be used first. So, the Lexicon token has the highest priority, followed by the different format tokens, and lastly the String token. During our development, we found that many abbreviations are used within company names. This results in the fact that many capitalized or upper-case strings contain "dot" signs in them, e.g., "H.K.". Therefore, our implementation of those tokens deliberately allow the existence of some prunctuations (in particular ".").

For example, the job ad segment

"Fortis Clearing (OPTIONS) Hong Kong Ltd (a member of ..."

can be tokenized as:

"nil ** (Cap)(Cap) (Upper) (Cap)(Cap) (<Company>) ** (Prun) "

where the two "**" indicate the boundaries between the target pattern and its neighborhood. Here we assume that only one item is to be extracted from the neighborhood (single-slot rules).

3.1.2 Initial Rule Set Generation

With all the items being tokenized in each sample, an intermediate format rule representation ($\hat{\mathbf{b}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{E}}$, $\hat{\mathbf{e}}$, $\hat{\mathbf{g}}$) is constructed by going through the following sequence of steps:

- "$\hat{\mathbf{b}}$": Scan from the left boundary towards left and stop just before the token type changes. Count the corresponding number of token instances encountered to compute the occurrence parameter of that token.
- "$\hat{\mathbf{B}}$": Scan from the left boundary towards right and repeat the same process.
- "$\hat{\mathbf{E}}$": Scan from the right boundary towards left and stop just before the token type changes or when a previously scanned item is encountered. Again, compute the occurrence parameter if needed.
- "$\hat{\mathbf{e}}$": Scan from the right boundary towards right and again stop just before the token type changes. Also, compute the occurrence parameter by counting.
- "$\hat{\mathbf{g}}$": Count the items between $\hat{\mathbf{B}}$ and $\hat{\mathbf{E}}$ which have not been scanned.

Thus, the rule generated for the above job ad segment is

" $\hat{\mathbf{b}}$=nil, $\hat{\mathbf{B}}$=(Cap,2), $\hat{\mathbf{E}}$=(<Company>), $\hat{\mathbf{e}}$=(Punct,1), $\hat{\mathbf{g}}$=3."

See Figure 1 for a pictorial illustration. Note that we use "()" instead of "[]" to indicate that it is an intermediate rule representation (with exact-length tokens) where the occurrence parameters of the tokens are interpreted as the exact counts, instead of the upper bound as defined in Section 2.1.

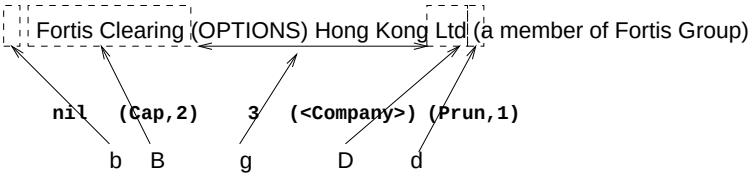


Figure 1. An example of intermediate format rule representation.

Lastly, the words associated with each token are added to complete the initial rule set generation, as shown in Figure 2.

$\hat{b}$	$\hat{B}$	$\hat{E}$	$\hat{e}$	$\hat{g}$
Format: nil List: nil	Format: (Cap,2) List: Fortis, Clearing	Format: (<Company>) List: nil	Format: (Punct,1) List: (	3

Figure 2. The Complete Rule Representation

3.1.3 Rule Generalization

After the inital rule set generation step, each training sample generates exactly one rule with obviously limited generalization capability. A bottom-up covering process (similar to the one used in [1]) is then applied to generalize the rules. The basic idea is to first group rules based on the token types to form a number of disjoint rule subsets. For each subset, we generalize the rules in it in a pair-wise manner with the goal of forming a rule set covering the largest number of examples without extraction errors. For example, the rule shown in Figure 2 belongs to the rule subset "`nil#Cap#<Company>#Punct`". During the generalization process, a pair of rules from it will be randomly picked. and generalized to a set of less specific rules. The maximum occurrence parameter of the rule pair is used in the newly generated rules. For example, if "Cap,1" is the beginning token of the first rule and "Cap,3" is that of the second, the final beginning token will be "Cap,3". We treat the maximum number of strings allowed parameter (**g**) in a similar manner. For the associated word lists, we combine the word lists of the two rules to form a new rule, and drop the word lists to form another.

These newly generated rules will not be adopted at once. They need to be tested with the training set to see if they can match all training data correctly. To avoid the rules to be too specific to the training data (overfitting), a certain degree of matching errors (later on called the *tolerated error rate*) is allowed. So, each new rule is first matched with the training data and the numbers of correct and failed matches are recorded. Here, we consider correct company name extractions as well as empty extractions from ads without company names to be correct matches. All the other cases are considered as failed. If the percentage of failed matches exceeds the tolerated error rate, the new rule will be discarded.

New rules that can survive will then undergo a *subsuming* process. Subsuming is a process that replaces specific rules by more generalized rules, i.e. can cover more data. For example, if there are two rules with identical format, the rule without a word list will be more general the one with a word list. In other words, the former rule subsumes the latter one and thus the latter can be removed. If both rules have a word list with them, the latter rule's word list should contain all the words in the former rule's word list to perform subsuming. After subsuming of the old rules in the group, the new rules will finally be added into the group. This generalization process iterates until there is no more change in the number of rules in the group for a certain number of iterations. Figure 3 summarizes the overall learning process.

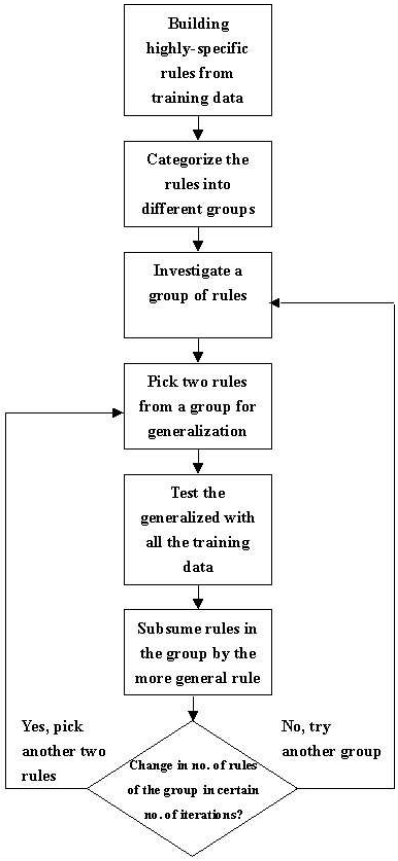


Figure 3. The whole process of rule generalization.

We believe that such a generalization process can eliminate rules with tokens carrying a big trunk of words as far as possible and retain those with tokens carrying a short list of important keywords.

3.2 Pattern Extraction

After rule learning, the resultant set of rules forms the representation of the target pattern. By applying the rules to an unknown input document of the same application domain, the system is expected to be able to extract all the relevant patterns out and only the relevant patterns. Any spurious extraction results should be considered as failure cases. It is also expected that the order of the rule application should be independent to the extraction results, i.e. the extraction results will not be affected by the order of applying the rules.

4 Experiments

A prototype of the proposed system has been implemented in Perl 5. To evaluate the system performance, we have created a dataset by retrieving job ads from a local online classified post and preprocessed them so that irrelevant information like company logos, graphical ad banners, etc. are removed. A total of 561 job ads which have been preclassified into 22 job categories, are solicited to form our dataset. Among the 561 job ads, only 241 of them contain company names.

4.1 Preprocessing

All the job ads in the dataset are in HTML format. As the core information of the job ad is found to be presented in a consistent manner (embedded inside a particular table), the preprocessing step is as simple as writing some simple pattern matching rules based on HTML tags. In general, this very first extraction step may not necessarily be that simple [3].

4.2 Extraction of Company Names

The proposed system has been applied to extract company names from the preprocessed job ads as described in Section 3.2. The labeled data are divided into the training set and the test set and cross-validation has been used to reduce the sampling bias. The rules are learned based on the training set and applied to the test set for evaluation. In order to demonstrate the effect of different rule representations and the advantage of using the induction process for rule creation, we set up four different experiments. In Experiment 1, we use the rule representation that contains only token classes but not word lists. Note that rule induction is not needed in this experiment. In Experiment 2, we allow the use of word lists in the prefix and suffix sub-patterns only. In Experiment 3, we add word lists in the target pattern representation instead. In both Experiment 2 and 3, the rule generalization process as described in Section 3.1.3 is used. Lastly, in Experiment 4, we allow the use of word lists in all parts of the rules and rely on the system to keep or drop the word lists or the token classes in both the prefix/suffix sub-patterns as well as the target patterns. Performance comparison based on the extraction accuracy is shown in Table 1, 2, 3 and 4 (also revealed in Figure 4). According to Figure 4, it is first observed that as the size of training set increases, the extraction accuracy increases monotonically in all the experiments. For the comparison of different rule representations, we see that introducing word lists in the prefix and suffix sub-patterns results in significant performance degradation (Experiment 2). This signs that including word lists only in the representation of the prefix and suffix sub-patterns fails to enhance the model accuracy.[2] However, significant performance improvement is achieved by adding word lists to the target pattern representation instead (Experiment 3). Lastly, the best performance is achieved by allowing word lists in all parts of the rule representation (Experiment 4). This evidences the system's capability of finding the optimal rule representation automatically. Other than comparing different representations, we have also evaluated the use of the tolerated error rate in enhancing the system's generalization performance. As described in Section 3.1.3, setting the tolerated error rate to $r\%$ means that an induced rule is admitted into the rule set as far as it makes less than $r\%$ extraction errors on the training set. In particular, when $r = 0$, all the induced rules are not allowed to make any error at all on the training set, and in general cannot result in optimal generalization performance. Figure 5 shows the performance comparison based on different values of r. We note that the use of the tolerated error rate always helps when the number of training data is significantly small (see the performance when the number of training data is 35). Also, we found that consistant improvement can be obtained by allowing a small degree of errors (tolerated error rate = 0.2).

# Training	35	49	81	161	193	207
Accuracy(%)	71.03	72.17	77.24	82.79	82.95	83.72

Table 1. Extraction results of experiment 1 where only typesetting information is used for the rule representation (tolerated error rate=0).

# Training	35	49	81	161	193	207
Accuracy(%)	68.76	70.38	74.20	79.76	79.90	80.48

Table 2. Extraction results of experiment 2 where typesetting information and word lists (limited to prefi x and suffi x) are used for the rule representation. (tolerated error rate=0)

# Training	35	49	81	161	193	207
Accuracy(%)	67.41	72.09	75.90	87.83	89.76	91.23

Table 3. Extraction results of experiment 3 where typesetting information and word lists (limited to company name) are used for the rule representation (tolerated error rate=0)

[2] At least, our existing induction process fails to learn a representation which can perform better than the one using only typesetting information in the representation.

# Training	35	49	81	161	193	207
Accuracy(%)	68.46	72.45	76.80	88.73	91.02	92.13

Table 4. Extraction results of experiment 4 where both typesetting information and word lists are used for the rule representation. (tolerated error rate=0)

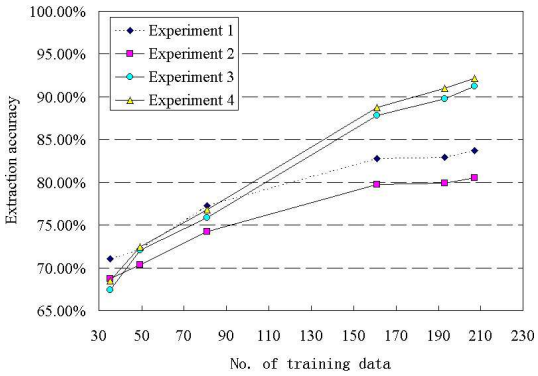


Figure 4. Comparing the performance of three different rule representations given the tolerated error rate = 0.

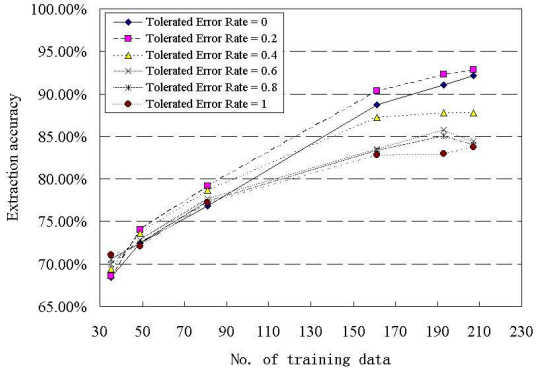


Figure 5. Comparing the performance for different values of tolerated error rates. Both word lists and token classes are used here for the rule representation.

5 Conclusion

In this paper, we have described an information extraction system we developed and show that our approach works well in company name identification in job ads (with an accuracy of 92.13%). We believe that extending it to extract job titles should be straight forward as job titles are normally highlighted in jod ads. However, when dealing with target items related to job natures and job requirements, typesetting matching will not be useful. Also, rule-based approaches are falling short in evaluating the strength of the evidence that guides extraction decisions [7]. More robust extraction technique are required. Currently, we are investigating the use of hidden Markov models to extract the fields other then company names. Also, coordinating the individual field wrappers to further improve the extraction accruacy is another worth-pursuing direction.

References

[1] M. E. Califf and R. J. Mooney. Relational learning of pattern-match rules for information extraction. In *Working Notes of AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, pages 6–11, Menlo Park, CA, 1998. AAAI Press.

[2] M.E. Califf. *Relational Learning Techniques for Natural Language Information Extraction*. PhD thesis, Department of Computer Science, University of Texas, Austin, TX., 1998.

[3] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.K. Ng, and R.D. Smith. Conceptual-model-based data extraction from multiple-record web pages. In *Data and Knowledge Engineering*, November 1999.

[4] D. Freitag. Information extraction from HTML: Application of a general machine learning approach. In *Proceedings of the Fifteenth National Conference on Articial Intelligence*, pages 517–523, 1998.

[5] C.N. Hsu and M.T. Dung. Generating finite-state transducers for semi-structured data extraction from the web. *Information Systems*, 23(8):521–538, 1998.

[6] N. Kushmerick, D.S. Weld, and R. Doorenbos. Wrapper induction for information extraction. In *Proceedings of International Joint Conference on Artificial Intelligent*, 1997.

[7] Nicholas Kushmerick. Finite-state approaches to web information extraction, 2002.

[8] I. Muslea. Extraction patterns for information extraction tasks: A survey. In *Proceedings of AAAI-99 Workshop on Machine Learning for Information Extraction*, Orlando, Florida, July 1999.

[9] I. Muslea, S. Minton, and C.A. Knoblock. Hierarchical wrapper induction for semistructured information sources. *Journal of Autonomous Agents and Multi-Agent Systems*, 2000.

[10] S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, pages 1–44, 1999.

Distributed Task Assignment for Information Gathering

Katsutoshi Hirayama
Kobe University of Mercantile Marine
5-1-1 Fukae-minami-machi, Higashinada-ku,
Kobe 658-0022, JAPAN
hirayama@ti.kshosen.ac.jp

Yasuhiko Kitamura
Graduate School of Engineering, Osaka City University
3-3-138 Sugimoto-cho, Sumiyoshi-ku,
Osaka 558-8585, JAPAN
kitamura@info.eng.osaka-cu.ac.jp

Abstract

This paper describes a method to solve the distributed task assignment problem on a simple model of multi-agent information gathering. This model consists of two types of agents: Information Integration agents (I-agents) and Information Gathering agents (G-agents). I-agents assign tasks of watching some information sources to G-agents and G-agents notify I-agents of renewal of information sources that they have been watching. In this model, I-agents need to assign watching tasks so that the assignment meets not only users' requirements but also resource constraints imposed upon G-agents. Also, it is desirable that I-agents can obtain such an assignment without revealing users' requirements each other because such requirements may include some private information. In this paper, we encode this assignment problem as distributed SAT and solve it using a general-purpose distributed SAT algorithm.

1. Introduction

The Web on the Internet is now widely used as a basic tool for broadcasting information worldwide. Since anyone in the world can easily broadcast his/her messages through the Web, the Web almost behaves like a distributed database which covers various topics and is continuously updated without global control. However, it is very difficult for users to gather useful information effectively on such a huge and dynamic distributed database. In this work, we aim to develop a simple model of multi-agent information gathering and design a tool that can support an information gathering process on a huge and dynamic distributed database.

In this work, we assume that an information gathering process consists of the following two simple tasks:

- determining which information source should be watched
- watching the information source and notifying someone if it is updated

Performing these tasks on a huge and dynamic distributed database naturally gives us an idea of modeling this process as an interaction among agents. In this work, we introduce *Information Integration agents* (I-agents) that perform the first task and *Information Gathering agents* (G-agents) that perform the second task and realize information gathering on a huge and dynamic distributed database through an interaction among I-agents and G-agents. We illustrate the model in Fig. 1. A typical scenario of the information gathering process in this model is as follows.

1. I-agents assign information sources to G-agents. I-agents must find an appropriate assignment considering users' requirements and resource constraints imposed upon G-agents.
2. When assigned information sources, G-agents slot them in their watching schedules to watch them periodically.
3. When finding the renewal of information sources, G-agents notify the corresponding I-agents of the fact.
4. When notified of the renewal of information sources, I-agents perform some procedures like notifying users or altering information sources to be watched.

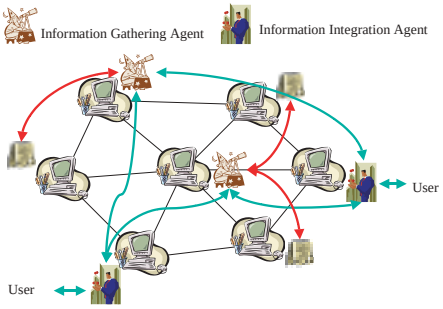


Figure 1. The Model

This paper focuses on the first step of the scenario, i.e., how I-agents assign information sources to G-agents so that both users' requirements and G-agents' resource constraints are met.

One solution to this problem is that I-agents put all users' requirements and all G-agents' resource constraints together to solve the entire problem by some centralized assignment algorithm. However, we consider that this solution is not appropriate in terms of security or privacy because users' requirements, which might include some confidential information (like personal interests in particular information sources), would be exposed to others. Therefore, we seek for a distributed solution where I-agents search for an appropriate assignment without putting their requirements and constraints together.

A simple distributed solution might be greedy search where an I-agent selects one available G-agent and assigns it an information source without coordinating with other I-agents. Although this method is very simple, it is easy to imagine that such a greedy method usually fails to obtain an assignment with good enough quality. In this paper, we provide a distributed solution to the assignment problem by encoding the problem as distributed SAT and solving it using a general-purpose distributed SAT algorithm.

The paper is organized as follows. First, we provide definitions of SAT and distributed SAT (Section 2). Then, we describe a way to encode the assignment problem as distributed SAT (Section 3). Next, we explain a distributed SAT algorithm called MULTI-DB (Section 4) and show the result of the experiment on simple examples (Section 5). Finally, we conclude this work and show some future work (Section 6).

2. Distributed SAT

Propositional satisfiability (SAT) is the problem of deciding if there is an assignment for variables in a propositional formula that makes the formula true. SAT has attracted considerable attention recently within the AI community. A lot of work has been made on SAT including efficient SAT solvers like GSAT[7] and WalkSAT[6], complexity analyses using the notion of "phase transition"[3], efficient SAT-based planning algorithms[2], and so on.

A propositional formula in conjunctive normal form (CNF formula) is a conjunction of clauses, where a *clause* is a disjunction of literals and a *literal* is a propositional variable or its negation. The following formula over the variables $\{x_1, x_2, x_3, x_4\}$ is an example of a CNF formula.

$$(x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_4) \wedge (\neg x_3 \vee \neg x_4) \\ \wedge (\neg x_1 \vee \neg x_3) \wedge (\neg x_2 \vee \neg x_4) \quad (1)$$

The truth assignment $(x_1, x_2, x_3, x_4) = (\mathrm{T}, \mathrm{F}, \mathrm{F}, \mathrm{T})$ is a solution of the formula (1) since it makes the formula true. We may note, in passing, that the formula (1) represents an example of 2-coloring problem with a graph that consists of two nodes linked each other.

Distributed SAT (DisSAT) is a problem where variables and clauses in a CNF formula are distributed among multiple agents. We usually assume that variables and clauses are distributed such that:

- Variables are partitioned into multiple agents, i.e., no variable is shared among agents.
- Each clause is assigned to all of the agents involved with the clause. In other words, an agent has all of the clauses relevant to its assigned variables.

A solution to DisSAT is the state where every agent finds truth values to its assigned variables that satisfy all of its assigned clauses.

Take CNF formula (1), for example. This formula consists of four variables and six clauses (we refer to them as $C_1, \ldots, C_6$). If there exist two agents, say $Agent_1$ and $Agent_2$, where each is assigned $\{x_1, x_2\}$ and $\{x_3, x_4\}$, respectively, then $Agent_1$ has $\{C_1, C_2, C_5, C_6\}$ and $Agent_2$ has $\{C_3, C_4, C_5, C_6\}$ as shown in Fig. 2.

In this case, $\{C_5, C_6\}$ are assigned to both agents because each of them includes both agents' variables. Clauses that include the variables of multiple agents, such as C_5 and C_6, are called *inter-agent clauses*. On the other hand, clauses that include only the variables of one agent, such as C_1, C_2, C_3, and C_4, are called *intra-agent clauses*. An agent (say a) usually has both inter- and intra-agent clauses and each of a's inter-agent clauses includes some *external variables* that belong to other agents. We call agents that a's external variables belong to a's *neighboring agents*. We

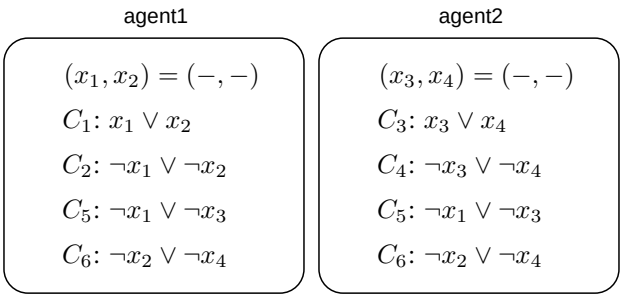


Figure 2. Example of DisSAT

refer to the set of all of a's neighboring agents as a's *neighbors*.

3. Encoding

The assignment problem we solve is how I-agents assign information sources to G-agents so that both users' requirements and G-agents' resource constraints are met. In this paper, we assume that both users' requirements and G-agents' resource constraints are described in very simple form.

A user's requirement takes the form of "want x to watch i", where x is a name of a specific G-agent and i is an information source that a user wants x to watch. A special form, "want any to watch i", is provided when a user wants any agent to watch i.

As resource constraints, we assume that the maximum number of information sources that a G-agent can take is given. Namely, a resource constraint on G-agent x is like "among possible information sources, x can watch up to 3 information sources". We also assume that the numbers of possible information sources for G-agents are finite.

Although these descriptions of users' requirements and G-agents' resource constraints are very simple, we believe that they take the essence of constraints in a assignment problem.

Now we can encode the assignment problem by I-agents as DisSAT. The way of encoding is as follows.

Agent: We regard I-agents as agents in DisSAT.

Variables: For any combination of I-agent a, G-agent x, and information source i, we introduce a propositional variable x_i^a whose value is true *iff* a assigns i to x.

Clause Set 1: Users' requirements received by I-agent a are encoded as a set of clauses $CS1^a$. For example, if a receives a request saying "want x to watch i" from a user, $CS1^a$ must include the unit clause x_i^a. Another example is: if a receives a request saying "want any to watch i" and knows that three G-agents, x, y, and z, can possibly watch i, $CS1^a$ must include the clause $x_i^a \vee y_i^a \vee z_i^a$.

Clause Set 2: I-agent a's knowledge of resource constraints on G-agents is encoded as a set of clauses $CS2^a$. For example, if a knows that G-agent x can possibly watch the information sources i, j, and k, but it can actually watch at most two sources among the possible sources, then $CS2^a$ must include the clause $\neg x_i^a \vee \neg x_j^a \vee \neg x_k^a$.

Clause Set 3: A final assignment of information sources to G-agents must be agreed on among all I-agents. Accordingly, I-agent a generates a set of clauses $CS3^a$ meaning its final assignment should be identical to other agents' final assignments. For example, if I-agent a needs to agree with I-agent b on the assignment to G-agent x that can possibly watch i and j, $CS3^a$ must include the following four clauses: $\neg x_i^a \vee x_i^b$, $x_i^a \vee \neg x_i^b$, $\neg x_j^a \vee x_j^b$, and $x_j^a \vee \neg x_j^b$.

We show a simple example. We assume that there are two I-agents, a and b, and two G-agents, x and y. Each G-agent can possibly watch information sources 1, 2, and 3, but actually do at most two sources because of their resource constraints. We also assume that a receives requests saying "want any to watch 1" and "want any to watch 2" while b receives those saying "want any to watch 1" and "want any to watch 3". A SAT encoding of this example is as follows. Note that for each agent all of the clauses in three clause sets must be satisfied.

I-agent a: variables: $\{x_1^a, x_2^a, x_3^a, y_1^a, y_2^a, y_3^a\}$

$CS1^a$**:** $\{x_1^a \vee y_1^a, x_2^a \vee y_2^a\}$
$CS2^a$**:** $\{\neg x_1^a \vee \neg x_2^a \vee \neg x_3^a, \neg y_1^a \vee \neg y_2^a \vee \neg y_3^a\}$
$CS3^a$**:** $\{\neg x_1^a \vee x_1^b, x_1^a \vee \neg x_1^b, \neg x_2^a \vee x_2^b, x_2^a \vee \neg x_2^b, \neg x_3^a \vee x_3^b, x_3^a \vee \neg x_3^b, \neg y_1^a \vee y_1^b, y_1^a \vee \neg y_1^b, \neg y_2^a \vee y_2^b, y_2^a \vee \neg y_2^b, \neg y_3^a \vee y_3^b, y_3^a \vee \neg y_3^b\}$

I-agent b: variables: $\{x_1^b, x_2^b, x_3^b, y_1^b, y_2^b, y_3^b\}$

$CS1^b$**:** $\{x_1^b \vee y_1^b, x_3^b \vee y_3^b\}$
$CS2^b$**:** $\{\neg x_1^b \vee \neg x_2^b \vee \neg x_3^b, \neg y_1^b \vee \neg y_2^b \vee \neg y_3^b\}$
$CS3^b$**:** $\{\neg x_1^a \vee x_1^b, x_1^a \vee \neg x_1^b, \neg x_2^a \vee x_2^b, x_2^a \vee \neg x_2^b, \neg x_3^a \vee x_3^b, x_3^a \vee \neg x_3^b, \neg y_1^a \vee y_1^b, y_1^a \vee \neg y_1^b, \neg y_2^a \vee y_2^b, y_2^a \vee \neg y_2^b, \neg y_3^a \vee y_3^b, y_3^a \vee \neg y_3^b\}$

Assuming that the number of I-agents is m, the number of G-agents is n, the number of possible information sources for each G-agent is s, the maximum number of information sources that each G-agent can take is t, and the number of users' requirements that each I-agent receives is u. Then, the size of resulting SAT for each I-agent, say a, is: the number of variables is ns, the size of $CS1^a$ is u, the size of $CS2^a$ is $n \cdot {}_sC_{t+1}$, and the size of $CS3^a$ is $2ns(m-1)$.

```
(01) for t := 1 to MAXTRIES do
(02)   set variable values randomly;
(03)   for r := 1 to MAXROUNDS do
(04)     exchange variable values with neighbors;
(05)     perform local search for PossFlips;
(06)     exchange PossFlips with neighbors;
(07)     if there is no violated clause among a and its neighbors then
(08)       perform the termination detection procedure;
(09)     else
(10)       if there is no PossFlips among a and its neighbors then
(11)         increase weights of violated clauses;
(12)       else
(13)         for each newly violated clause at the next possible state do
(14)           if (the violation would be caused by at least two agents including a)
               ∧ (a's PossFlips would have the least improve among them) then
(15)             withdraw one of a's PossFlips that would cause the violation;
(16)           end if;
(17)         end do;
(18)         if no flip of a's PossFlips is withdrawn then
(19)           perform all the flips in PossFlips;
(20)         else
(21)           perform local search again for the background flips;
(22)           perform the background flips;
(23)         end if;
(24)       end if;
(25)     end if;
(26)   end do;
(27) end do;
```

Figure 3. MULTI-DB **(performance sketch for agent a)**

4. Algorithm

By the method described above, the assignment problem among I-agents is encoded as DisSAT and can be solved by a general-purpose DisSAT algorithm. In this work, we use MULTI-DB [1], which is an iterative improvement type of DisSAT procedure and capable of solving DisSAT where each agent has multiple local variables.

In MULTI-DB, agents iteratively improve their "flawed" variable values toward a solution by performing local search simultaneously while exchanging their (temporal) variable values each other. We show the performance sketch of MULTI-DB in Fig. 3. We briefly describe major steps of the performance sketch in this paper. Details of the procedure are shown in [1].

Each agent, say a, repeats a *try* with randomly chosen initial variable values until a specified upper bound, *MAXTRIES*, is reached. At each try, agent a randomly sets initial variable values and repeats from Step (04) through Step (25) until *MAXROUNDS* is reached. We outline the procedure from Step (04) through Step (25) below.

Steps (04) – (06) After exchanging variable values with its neighbors, agent a evaluates the current state as a weighted sum of violated clauses and searches for *PossFlips*, which is a set of possible variable flips that is able to reduce a weighted sum of violated clauses, by using a local search procedure. Then, agent a exchanges *PossFlips* with its neighbors.

Step (08) If there is no violated clause among a and its neighbors, agent a follows the termination detection procedure that is similar to the one in the distributed breakout[9].

Step (11) If there is at least one violated clause but no *PossFlips* among a and its neighbors, agent a gets stuck at a *quasi-local-minimum*[9]. When getting stuck at a quasi-local-minimum, agent a tries to escape from it by the *breakout strategy*[5], i.e., increasing the weights of violated clauses.

Steps (13) – (17) If agent a has *PossFlips* and there is at least one agent having *PossFlips* among its neighbors,

agent a needs to check if those *PossFlips* make them invalid each other before actually performing its *PossFlips*. To do this, agent a reasons the next possible state, where all *PossFlips* are performed, and identifies clauses that would newly get violated. Then, for each newly violated clause, agent a checks if 1) the new violation would be caused by at least two agents including a and 2) a's *PossFlips* would have the least improve in a weighted sum of violated clauses among the agents (ties are broken in favor of the agent with the larger ID); and if both are true, a withdraws one of the flips in its *PossFlips* that would cause the new violation.

Steps (18) – (23) If no flip is withdrawn in its *PossFlips* after the above procedure, it means that a's *PossFlips* can actually reduce a weighted sum of violated clauses even though its neighboring agents perform their *PossFlips*. Agent a therefore performs its *PossFlips* immediately. On the other hand, if some flips are withdrawn in its *PossFlips*, the remaining flips have to be reconsidered since a subset of *PossFlips* cannot always reduce a weighted sum of violated clauses. Accordingly, agent a performs local search again for the "background flips" that can be performed for the variables corresponding to the remaining flips, and then does the flips immediately.

MULTI-DB is incomplete, i.e., it may fail to find a solution even if a solution exists and cannot find the fact that no solution exists. However, the previous study shows that MULTI-DB is very effective for hard and satisfiable DisSAT[1]. Moreover, MULTI-DB is a "memory-saving" algorithm since it does not employ a memory-consumptive method like nogood learning, which plays an important role in the efficiency of the asynchronous type of distributed constraint satisfaction algorithms[8, 10]

5. Evaluation

As a first step toward an evaluation of our approach, we made an experiment on simple examples of the distributed task assignment problem using a simulator of a *fully synchronous distributed system*. A fully synchronous distributed system is a typical model of distributed system, where all agents synchronously repeat a cycle of the following activities: receiving all incoming messages, performing local computation, and sending messages. On this simulator, we implemented MULTI-DB and measured *cycles* and *flips* as its communication and computation costs, respectively.

Cycles is the number of cycles consumed until MULTI-DB finds one solution to a problem. Since all agents perform receiving all incoming messages, performing local computation, and sending messages in one cycle, the number of communication among agents increases with the number of cycles. Hence we consider *cycles* as the communication cost of MULTI-DB.

On the other hand, *flips* is the total sum of the maximal number of flips over the agents in each cycle until MULTI-DB finds one solution to a problem. More specifically, in each cycle, we identify the "bottleneck agent", which performed the most flips for its local computation, and sum up all of the maximal numbers of flips over all consumed cycles. Although the amount of computation in each cycle varies among the agents, the total amount of computation is dominated by the bottleneck agents. This measure can be thus considered as the computation cost of MULTI-DB.

In this experiment, we made the following three simple examples of the distributed task assignment problem.

Ex. 1 :

- I-agents: $\{a, b\}$, where a receives a request saying "wants any to watch 1 and 2" and b receives that saying "wants any to watch 2 and 3".
- G-agents: $\{x, y\}$, where each can possibly watch the information sources $\{1, 2, 3\}$, but actually do at most two sources.

Ex. 2 :

- I-agents: $\{a, b, c\}$, where a receives a request saying "wants any to watch 1, 2, and 3", b receives that saying "wants any to watch 2, 3, 4, and 5", and c receives that saying "wants any to watch 4, 5, and 6".
- G-agents: $\{x, y, z\}$, where x can possibly watch the information sources $\{1, 2, 3, 4\}$, y can possibly watch $\{1, 2, 5, 6\}$, and z can possibly watch $\{3, 4, 5, 6\}$. Each can actually watch at most two sources, however.

Ex. 3 :

- I-agents: $\{a, b, c\}$, where a receives a request saying "wants any to watch 1 and 2", b receives that saying "wants any to watch 3 and 4", and c receives that saying "wants any to watch 5 and 6".
- G-agents: $\{u, v, w, x, y, z\}$, where each can possibly watch the information sources $\{1, 2, 3, 4, 5, 6\}$, but actually do at most one source.

In the above examples, each agent has 6 variables and 16 clauses for Ex. 1, 12 variables and 63 or 64 clauses for Ex. 2 (b has 64 clauses while the others have 63 clauses), and 36 variables and 236 clauses for Ex. 3.

Table 1. Results of MULTI-DB for Ex. 1, 2, and 3

	cycles			flips		
	mean	median	stdev.	mean	median	stdev.
Ex.1	6.0	6.0	1.41	21.0	20.0	6.24
Ex.2	32.7	30.0	18.5	167.3	160.5	96.7
Ex.3	89.2	90.0	48.0	461.3	470.5	246.3

Table 1 indicates the result of the experiment. Note that for each example we generated 50 sets of initial variable values and made MULTI-DB run with each set, i.e., we made 50 runs for each example. We show the mean, median, and standard deviation of cycles/flips over those 50 runs in Table 1. Also note that in MULTI-DB we set *MAX-TRIES* to 1 and *MAXROUNDS* to 1000 and used a variant of WalkSAT[6] with the tabu list (noise = 0.3, tabu length = 5) as a local search method.

Although this experiment is not a comparative study and hence we cannot lead to clear conclusions, we have a feeling that MULTI-DB can work fairly well for the distributed task assignment problem. Even for Ex. 3 encoded as DisSAT with 108 variables and 492 clauses (where an inter-agent clause, whose copies are distributed to multiple agents, is counted as one clause) in total, the costs of MULTI-DB are not so large. We should note that for the uniform random 3-SAT in SATLIB (http://www.satlib.org/benchm.html) with similar sizes, the mean cycles of MULTI-DB are over 500[1]. We suppose that this efficiency may come from the property of a problem. Indeed, for the DisSAT of Ex. 3, 486 out of 492 clauses are two-length clauses (clauses that consist of only two literals). It is well known that 2-SAT, which consists only of two-length clauses, is in class P.

Obviously, this experiment is very preliminary. That is partly because the encoding a problem as DisSAT was done manually in this experiment. Our future work will include detailed experiments that might reveal the hardness/easiness of the distributed assignment problem.

6. Conclusions and Future Work

We described a method to solve the distributed task assignment problem on a simple model of multi-agent information gathering. Our approach is that we first encode the distributed task assignment problem as distributed SAT and then solve it using a general-purpose distributed SAT algorithm called MULTI-DB.

A major contribution of this paper is that we showed how the distributed task assignment problem is encoded as DisSAT. We believe that this work is important especially for the distributed constraint satisfaction community since the issue of describing a realistic problem as DisCSP has not been fully investigated (excepting [4], [11], and etc.).

Obviously, there remain lots of future work: conducting detailed experiments on the complexity of the distributed task assignment problem, testing the performance of other DisSAT/DisCSP algorithms for this problem, and so on.

References

[1] K. Hirayama and M. Yokoo. Local search for distributed SAT with complex local problems. *Proc. of the First International Joint Conference on Autonomous Agents & Multi-Agent Systems*, pages 1199–1206, 2002.

[2] H. A. Kautz and B. Selman. Pushing the envelope: Planning, propositional logic, and stochastic search. *Proc. of AAAI-1996*, pages 1194–1201, 1996.

[3] D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of SAT problems. *Proc. of AAAI-1992*, pages 459–465, 1992.

[4] P. J. Modi, H. Jung, M. Tambe, W.-M. Shen, and S. Kulkarni. A dynamic distributed constraint satisfaction approach to resource allocation. *Proc. of the Seventh International Conference on Principles and Practice of Constraint Programming*, pages 685–700, 2001.

[5] P. Morris. The breakout method for escaping from local minima. *Proc. of AAAI-1993*, pages 40–45, 1993.

[6] B. Selman, H. A. Kautz, and B. Cohen. Noise strategies for improving local search. *Proc. of AAAI-1994*, pages 337–343, 1994.

[7] B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. *Proc. of AAAI-1992*, pages 440–446, 1992.

[8] M. Yokoo, E. H. Durfee, T. Ishida, and K. Kuwabara. The distributed constraint satisfaction problem: formalization and algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 10(5):673–685, 1998.

[9] M. Yokoo and K. Hirayama. Distributed breakout algorithm for solving distributed constraint satisfaction problems. *Proc. of the Second International Conference on Multi-Agent Systems*, pages 401–408, 1996.

[10] M. Yokoo and K. Hirayama. Distributed constraint satisfaction algorithm for complex local problems. *Proc. of the Third International Conference on Multi-Agent Systems*, pages 372–379, 1998.

[11] W. Zhang and Z. Xing. Distributed breakout vs. distributed stochastic: A comparative evaluation on scan scheduling. *Proc. of the Third International Workshop on Distributed Constraint Reasoning*, pages 192–201, 2002.

A Novel Incremental SVM Learning Algorithm

Ma Jian[1] Zeng Wenhua[2]
[1]Hangzhou Institute of Electronics Engineering, Hangzhou, P.R.China, 310037
Majian1979@msn.com
[2]Xiamen University, Xiamen, P.R.China, 361005
zengwenhua1964@hotmail.com

Abstract

Incremental Learning Technology is very important to extract valuable knowledge from many large dataset in different domain. Support Vector Machine (SVM) as a promising machine learning technique has been successfully applied to incremental learning by some researchers. This paper presents a novel incremental SVM learning algorithm based on previous works. The novel algorithm is derived from the analysis of the relation between incremental samples and Kuhn-Tucker condition of the old SVM obtained from old samples. In training process certain old samples which violate Kuhn-Tucker condition of new SVM classifier derived from incremental samples, are retrained, compared with the previous works that only concern the incremental samples. The resemblances and differences between our algorithm and previous works are discussed. Analysis suggests that our algorithm is a lossy approximation of Osuna's Decomposition method. Experimental results show that our algorithm is advantageous than previous similar works.

1. Introduction

Knowledge contained in many large data sets from many domains should be extracted in that they are very helpful to know the things and make decision for the future. Researchers in the machine learning and data mining have therefore been contributing to the solution of this problem. Learning based on examples usually need all data before it begin. Unfortunately, large data sets are too big to be loaded into memory at one time. To overcome this problem, incremental learning technique is presented.

Incremental learning technology has been widely researched because it can not only discard some useless samples to reduce the burden of memory, but also make full use of historical learning results successively.

Many incremental learning algorithms based on traditional learning techniques have been presented.[1,2] Traditional machine learning techniques are lack of mechanics which can ensure good generalization, so algorithms based on traditional methods often tend to be over-fitting or local minimal.

Compared with traditional learning techniques, Support Vector Machine (SVM) that is developed by Vapnik and his colleges [3,4,6], due to its strong learning ability and good generalization ability has been considered as a promising method on incremental learning techniques. Some algorithms based on SVM have been presented [9,10] and show good learning results on different datasets.

In this paper, we present a novel incremental SVM learning algorithm. First, we discuss the relation between samples and Kuhn-Tucker (KT) condition of old SVM classifier. The effect of incremental samples on incremental learning result is discussed. According to discussion result, a novel incremental learning algorithm is presented that is based on some previous works.

This paper is organized as follows: in the next section, we give a brief view on SVM. In section 3 we discuss the relation between samples and KT condition of old SVM classifier, and also possible change of support vectors set. We present our new incremental learning algorithm in section 4. In section 5, we compare our algorithm with some previous works. In section 6, we give the experiment results of the above algorithm. Finally, in section 7, the conclusion and further research are discussed.

2. Support Vector Machine

Standard SVM can solve a two-class classify problem with sample set $\{X_i, y_i\}$ $i = 1, \cdots, l$, $X_i \in R^n$, $y_i \in \{\pm 1\}$. The basic idea of it is finding a hyperplane that can group samples from the same class into the sample side of the plane, and at the same time maximizing the margin between classes to ensure a good generalization.

Obtaining such a classifier is equal to solving an optimal problem: goal of separating samples into two parts gives constraint conditions, maximizing the margin makes cost function of optimal problem. Let the hyperplane equation be: $wX + b = 0$ (1), then optimal problem is a quadratic one:

$$\min \frac{\|w\|^2}{2} + C\sum_{i=1}^{l}\xi_i \qquad (1)$$
$$\text{s.t.} \begin{matrix} y_i(wX_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{matrix}$$

The dual problem be:

$$\min : W = \frac{1}{2}\sum_{i,j}\alpha_i Q_{ij}\alpha_j - \sum_i \alpha_i + b\sum_i y_i\alpha_i \quad (2)$$

with Lagrange multipliers $0 \leq \alpha_i \leq C$, plane offset b and $i = 1, \cdots, l$, with symmetric positive definite kernel matrix $Q_{ij} = y_i y_j K(X_i, X_j)$. Classifier function be:

$$f(X) = \text{sgn}(\sum_{i=1}^{l} a_i y_i K_i(X_i, X) + b). \qquad (3)$$

3. Analysis On Incremental Learning

Kuhn-Tucker Condition of SVM

Kuhn-Tucker (KT) condition of the dual problem is:

$$\frac{\partial W_i}{\partial \alpha_i} = y_i f(X_i) - 1 \begin{cases} \geq 0 & \alpha_i = 0 \\ = 0 & 0 < \alpha_i < C \\ \leq 0 & \alpha_i = C \end{cases} \qquad (4)$$

$$\frac{\partial W_i}{\partial b} = \sum_i y_i\alpha_i = 0$$

According to KT condition, after the problem is solved, the distribution of samples is as follows:

Samples associated with $\alpha = 0$ distribute out of the margin, those associated with $0 \leq \alpha \leq C$ are on the boundary of margin, and those with $\alpha = C$ are in the margin as shown in Figure1.

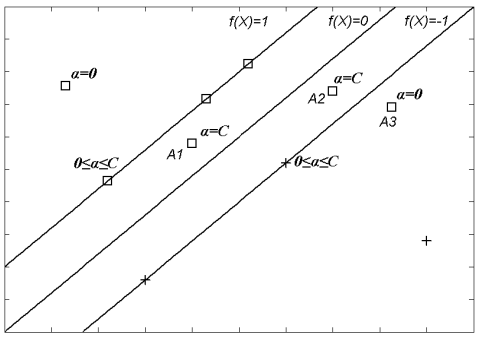


Figure 1. Relation between distribution of samples and KT condition of old SVM classifier.

What incremental learning algorithm concerns is how new knowledge can be extracted from incremental samples. To extract such knowledge means to revise SVM classifier to adapt to incremental samples. In the following section, our discuss will mainly focus on the following questions: what kind of incremental samples set will change the SV set? How the change will happen? What the new SV set will consist of?

Given an SVM classifier obtained from samples $\{X_i, y_i\}$, $i = 1, \cdots, l$ with classify function as (3), I is incremental samples $\{X_j, y_j\}$, $j = 1, \cdots, l_I$, l_I is the size of incremental set. Samples contain new information of classify means they can lead to more smaller value of (2). It also means they violate KT condition given by (4). Under this case, the old hyperplane and margin is not the best we can obtain from the whole sample set. Incremental samples have not been considered in the previous optimal process, so Lagrange multiplier associated with them is 0. If they satisfy KT conditions of SVM it means that $y_i f(X_i) - 1 \geq 0$ is true. On the contrary, if samples violate KT condition it can be de-rived that

$$y_i f(X_i) - 1 < 0 \Rightarrow \begin{cases} -1 < y_i f(X_i) < 1 \\ y_i f(X_i) < -1 \end{cases} \qquad (5)$$

This suggests that samples that violate KT condition are made up of two parts: the first is samples in the margin, and the second is samples fell into the other class space that should be classified incorrectly. Furthermore samples in the margin between classes can be divided into two groups by $f(X) = 0$: those that can be classified correctly by the old SVM classifier, and those which should be misclassified. The first group samples of them satisfy $0 \leq y_i f(X_i) \leq 1$; and the second group samples satisfy $-1 \leq y_i f(X_i) \leq 0$. Two of the above total three groups are misclassified by the old SVM classifier. One of them is classified correctly. This suggests that on the incremental SVM learning problem, KT condition is more reasonable criteria than classifier function that is used to get useful samples that contain new knowledge. Only those violating KT condition can have influence on the incremental learning so on one hand samples satisfying KT condition can be discarded because classification information has been learning and has been contained in the support vectors set of the old SVM classifier, on the other hand samples violating KT condition should be added to the new incremental problem.

A Special Type of Samples

It must be paid attention to that if samples in the incremental set violate KT condition then some samples in the old samples might become support vector possibly. This can be proved by a special instance given in Figure 2.

The original support vector set is made up of S1-S5. A1-A3 are new samples. After training on the new samples, classify function $g(X)=0$ is obtained. It can be judged intuitively from figure 2 that S1, A3 and N1, which is an old sample, compose of a new support vector set.

This type of samples is a big trouble on the way of incremental SVM learning problem for in order to solve the problem, we cannot only consider samples violating KT conditions in the incremental samples set but also those old samples which are mentioned above.

One of basic idea of previous incremental learning is that incremental samples are incremental samples. That means that the final Learning result is a revision of previous learning result on the condition of the incremental samples. In fact, we not add the incremental samples to the old samples set but put them together. On the issue of incremental learning, each of them is as important to the problem as the other though the old samples set is generally bigger than the incremental one. Therefore if we look incremental samples set as the old samples set and train a new SVM classifier on them then the old samples can be treated as incremental samples to the incremental samples. Thus some samples in old sample set might violate KT condition of the SVM classifier obtained from incremental samples. They will be samples in the old dataset which are possibly become support vectors.

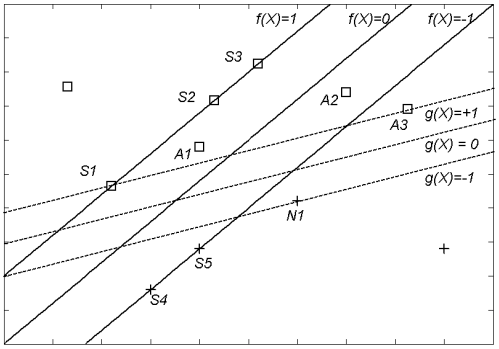


Figure 2. A special instance of old samples becoming support vectors in incremental learning.

4. Incremental Learning

A Incremental Learning Scheme

Based on the above analysis, we will present a new scheme of incremental learning in this section. First, a new SVM is derived from the incremental set. Then, samples in the old set, which violate KT condition of the new SVM and samples in incremental dataset which violate KKT condition of old SVM, and support vectors of both SVM classifier form a new training set. The SVM trained from such training set be the result.

In our scheme, not only classification error but also violation of KKT condition is the criteria. Both information in the old set and information, which have yet not been discovered in the new incremental set, are accumulated. Samples, which classified correctly by both SVM, are discarded as history and all the valuable samples are remained. Before we get the new smaller training set a train on incremental set is needed. And we validate on the whole set only once. Because incremental set is usually a small one and validation is an simple computation, the total computation burden of our scheme is very light.

It must be checked whether a sample violates KKT condition of some SVM. In fact, given the decision function of SVM classifier $f(X_i)$, $y_i \in \{\pm 1\}$, violating KKT condition has equivalence with $y_i f(X_i) < 1$. So we can present an incremental learning algorithm. Here two-class problem is considered because multi-classification can be transformed to many two-class problems.

Incremental Learning Algorithm

Presupposition: A SVM classifier Ω^0 is trained on the old sample set X_0. X_0^{SV} denotes support vector set of Ω^0, X_I is the incremental sample set.

Algorithm:

First, validate samples in X_I whether violates KKT condition of Ω^0. If none, stop. Ω^0 is the learning result. Otherwise, according to the result of validation, X_I can be divided into X_I^V and X_I^S. X_I^V denotes the set of samples that violate KKT condition of Ω^0. X_I^S denotes the set of samples which satisfy KKT condition of Ω^0.

Second, a new SVM classifier Ω^I is trained on the incremental sample set X_I. X_I^{SV} denotes support vector set of Ω^I.

Third, validate samples of the old sample set whether violate KKT condition of Ω^I. If none, stop. Ω^I is the learning result; otherwise according to validation result. X_o can be divided into X_0^V and X_0^S. X_0^V denotes the set of samples that violate KKT condition of Ω^1; X_0^S denotes the set of samples which satisfy KKT condition of Ω^1.

Finally, X_U denotes $X_0^{SV} \bigcup X_I^{SV} \bigcup X_0^V \bigcup X_I^V$. A new SVM classifier Ω is trained on X_U. Ω is the learning result. $X_H = X_0^S \bigcup X_I^S$ will be discarded as history set.

5. Compared with Previous works

Nadeem et al[9] have presented a framework for incremental learning with SVM. They divide a huge database into many partitions. Training on each partition can derive a small fraction of training examples, support vectors as learning result. The unions of these support vectors and incremental samples compose of the next training set.

In that we deal with a big dataset one part after another it's obvious that our method is similar to Nadeem's. The difference between us is that Nadeem only make use of old support vectors, but in our method not only support vectors are used but some additional samples of one sample set derived by KT condition of the other SVM classifier are also taken into consideration. Furthermore, the new training set is support vectors of a new SVM classifier which is derived from samples selected by our method. Our method can get more precise result than Nadeem's because of the additional samples and learning result is also a succinct one because it is obtained by training on the selected samples.

Both of Nadeem and our methods can be considered similar to Osuna's "chunking" techniques which is employed to train SVM[7]. In Osuna's method the original QP problem is replaced by a sequence of smaller size-fixed subproblems whose training set is called working set. Samples violating KT condition are selected into working set to replace samples in working set in order to improve the cost function strictly.

Incremental learning is also a training process to obtain a SVM classifier which can classify the old and incremental samples correctly, and each partition of big dataset can be seen as subproblem of the whole optimal problem. According to Osuna's "Building up" and "Building down" theorems[7], samples which are selected in our method into the new training set can improve the cost function. Hence the incremental learning result is an optimal one. Due to missing some valuable samples in old sample set, Nadeem's methods cannot make the cost function of the whole problem fully improved. Compared with Nadeem's method, our method's result is gotten from a better optimal process. Combined the geometrical property of support vector sets and the analytical property of optimal problem, our method find out more possible samples that should be replaced into the subproblem. Those samples are equivalent to or contained by samples that are built up into subproblem in Osuna's methods because they all violate the KT condition of subproblem composed of the other sample set, and contribute to minimize the cost function.

Unlike Osuna's recursive process, our method only optimizes the subproblems once, so it's also a lossy approximation of Osuna's method but at least a more precise approximation one than Nadeem's method. There are researchers which presents incremental SVM learning algorithm [10]. In their method, incremental learning algorithm process samples one by one. Though they can give almost best solution of the problem, computation burden of them is much bigger than ours method in that we deal with the new samples by batch. Compared with Osuna's and other's methods[8,9], ours need no recursive steps and less computation burden. It is a compromise between speed and accuracy.When there are too many incremental samples, our method is more competent in the case than those methods.

6. Experiment and Discussion

To evaluate the effectiveness of our algorithm, we implement the algorithm and then compare it with Nadeem's method and Osuna's method on datasets obtained from UCI machine learning repository. The datasets used in experiments are listed in Table 1. LibSVM[6], a library for support vector machines which implement a simplest instance of Osuna's method is adopted to obtain results of Osuna's method.

The experiment is designed as follows: each dataset is divided into 10 parts arbitrarily. Then a part is treated as the origin problem; other parts are added to the origin problem incrementally. The result of Osuna's method is obtained by training on the whole dataset once. A comparision between the above three methods on the size of final support vector set and prediction accuracy on the whole training dataset is shown in Table 2 and Table 3.

Dataset	# of Examples	No. of Attributes
Live-Disorder	345	6
Ionosphere	351	34
Mush-room	8124	22

Table 1. The datasets used in experiments.

Dataset	Osuna	Nadeem	Ours
Live-Disorder	247	225	222
Ionosphere	143	125	125
Mush-room	1102	827	881

Table 2. The size of Support Vector sets of each learning method's final results.

Dataset	Osuna	Nadeem	Ours
Live-Disorder	71.59	65.22	68.41
Ionosphere	94.59	95.16	95.16
Mush-room	100	99.64	100

Table 3. The final prediction accuracy of learning methods' final results on the whole dataset.

There are three noticeable comparison that: first, on live-disorder our algorithm's support vector set is smaller than Nadeem's but is more accuracy than theirs; second, on Ionosphere both incremental learning algorithm are more accuracy than traditional one; third, though our algorithm obtain much smaller support vector set than traditional one, our prediction is also prefect just as traditional one dose. From the results in Table 2 and Table 3, we can learn that though the final support vector set of our SVM incremental learning algorithm is smaller, some-times much smaller than that of traditional learning algorithm, our algorithm can still obtain almost the same prediction accuracy as the traditional one considering that both of incremental learning algorithms are a lossy approximation of traditional one. Our algorithm extracts slightly more valuable samples than Nadeem's algorithm for the prediction accuracy of our algorithm is better than theirs.

7. Conclusion and Further Research

We discuss the relations between incremental learning algorithm and the old SVM classifier and what kind of incremental and old samples might become support vectors possibly. Based on the discussion results, we present a novel incremental SVM learning algorithm. In our algorithm, incremental samples are treated equally with old samples and hence more valuable samples can be obtained as final learning results. The algorithm is novel is because that in training process certain old samples which violate Kuhn-Tucker condition of new SVM classifier derived from incremental samples, are retrained, compared with the previous works which only concern the incremental samples. Theoretical analysis suggests that our algorithm is advantageous than some similar previous works. Experiments results show that our algorithm can achieve a good learning result as expected in our analysis.

In the paper, experiment is made on certain common machine learning datasets. More experiments should be made in the future to examine the effectiveness of our algorithm. Our algorithm trains on incremental sample sets coming one after another. In fact, when facing a huge sample set we also can divided it into some relative small parts and then learn from them parallelly. So we will focus on the application of our incremental learning algorithm on parallel learning.

Reference

[1] J. Ratsaby, "Incremental learning with sample queries", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.20, Aug, 1998, pp. 883-888

[2] Wang, E.H.-C, and A. Kuh, "A smart algorithm for incremental learning", International Joint Conference on Neural Networks, Vol.3, 1992, pp. 121-126

[3] V. Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag, New York, 1995.

[4] V. Vapnik. "The Nature of Statistical Learning Theory", Springer-Verlag, New York, 2000.

[5] Christopher J.C.Burges, A Tutorial on Support Vector Machines for Pattern Recognition. Kluwer Academic Publishers, Boston, 1998.

[6] Chih-Chung Chang, and Chih-Jen Lin, "LIBSVM: a Library for Support Vector Machines", Available: http://citeseer.nj.nec.com/chang01libsvm.html

[7] E.Osuna, R. Freund, and F.Girosi, "Improved training algorithm for support vector machines", In Proc. IEEE Workshop on Neural Networks for signal Processing,1997, pp 276-285.

[8] G.Cauwenberghs, and T. Poggio, "Incremental and decremental support vector machine learning", In Adv. Neural Information Processing, volume 13, MIT Press, 2001.

[9] N. Syed, H. Liu, and K. Sung, "Incremental learning with support vector machines", In Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI), 1999.

[10] L.Ralaivola and F. d'Alche-Buc, "Incremental Support Vector Machine Learning: a Local Approach", In Proc. of ICANN'01, Austria, 2001.

Mathematical and Simulation Model of Fault Tolerance Distributed Database Systems

Maciej Kiedrowicz
Cybernetics Department
Military University of Technology
00-908 Warsaw, 2 Kaliskiego
Poland
E-mail: kiedrowicz@isi.wat.waw.pl

Abstract

In the paper it is considered the support method for distributed database design process for computer information systems, which are exposed to intentional destruction of computer network components (for instance military applications). The problem of distributed database fragmentation, allocation and replication is considered.

It tries to obtain the solution as a matrix, which describes the idea of database fragmentation, allocation and replication. We have five criteria that estimate database fragmentation, allocation and replication solution: communication load of network, data file dispersal (distribution), response time, memory occupation and disturbance tolerance. The problem of finding the optimal database fragmentation, allocation and replication, for first four criteria, is the binary problem of mathematical programming. For the last criterion we should use simulation method for searching the best solution. The discrete – event simulator SRBD is presented as an example of this method.

1. Introduction

There are many well-known methods used in database system designing. Nowadays often and often we can meet in literature articles, which describe those issues in the light of distributed database. Main role in a process of distributed database designing is played by multi-criteria optimization and simulation methods. Especially it concerns the simulation, which is often the best way to check correctness and a usefulness of such designs. In military area where computer networks are exposed to intentional destruction the simulation method is the practically only way to achieve satisfactory solutions.

2. Mathematical model

2.1. Symbols:

Z - set of data file numbers;

W - set of network node numbers;

F_z - set of fragment numbers, which can be separated from data file with number z, $z \in Z$;

V_f^z - capacity (in bytes) of fragment with number f, which was separated from file with number z, $f \in F_z$, $z \in Z$;

Δ_f^z - capacity (in bytes) of redundancy related to fragment with number f, which was separated from file with number z, $f \in F_z$, $z \in Z$;

X - decision variable vector, where:

$$X = (x^1, x^2, ..., x^z, ..., x^Z), \quad (1)$$

where: x^z - decision variable related to file with number z , $z \in Z$,

$$x^z = \left[x_{fw}^z\right]_{F_z x W}, \ z \in Z, f \in F_z, w \in W \quad (2)$$

where x_{fw}^z is equal one, when fragment with number f of file with number z was placed at node with number w and zero otherwise;

V^z - capacity of file with number z in bytes

$$V^z = \sum_{w=1}^{W} \sum_{f=1}^{F_z} x_{fw}^z \cdot V_f^z, \ z \in Z \quad (3)$$

$$V_\Delta^z = \sum_{w=1}^{W} \sum_{f=1}^{F_z} x_{fw}^z \left(V_f^z + \Delta_f^z\right), \ z \in Z, \quad (4)$$

where V_Δ^z - capacity of divided file with number z together with redundancy areas;

$\mathbf{d}$ - matrix defining length of roads between nodes.

2.2. The forms of criteria function are as follows:

- response time

$$F_1 = \frac{1}{Z} \sum_{z=1}^{Z} \sum_{f=1}^{F_2} \left(\sum_{j=1}^{W} \frac{h_{f,j}^z}{\sum_{i=1}^{w} h_{f,i}^z} \sum_{w=1}^{W} x_{fw}^z d_{jw} \right), \quad (5)$$

- memory occupation

$$F_2 = \sum_{w=1}^{W} \sum_{f=1}^{f_z} \sum_{z=1}^{Z} x_{fw}^{z} \left(V_f^z + \Delta_f^z \right), \quad (6)$$

- communicational load of network

$$F_3 = \frac{1}{U} \sum_{z=1}^{Z} \sum_{f=1}^{F_z} \left(\sum_{j=1}^{w} \frac{(h_{f,j}^{z})^2}{\sum_{i=1}^{w} h_{f,i}^{z}} \sum_{w=1}^{w} x_{fw}^{z} d_{jw} \right), \quad (7)$$

- data file dispersal (distribution)

$$F_4 = \sum_{z=1}^{Z} \sum_{w=1}^{W} \left[1 - \prod_{f=1}^{F_z} \left(1 - x_{fw}^{z} \right) \right], \quad (8)$$

- disturbance tolerance of computer network nodes – F_5; length of time interval in which database system is in working order; the value of the function is calculated by the simulation software SRBD.

2.3. Conditions for decision variables:

- decision variable is zero-one variable:

$$x_{fw}^{z} \in \{ 0, 1 \}, \; z = \overline{1, Z}, \; w = \overline{1, W}, \; f = \overline{1, F_z}, \quad (9)$$

- fragments in a node is not more than it is allowed due to capacity of the node

$$\sum_{z=1}^{Z} \sum_{f=1}^{F_z} x_{fw}^{z} (V_f^z + \Delta_f^z) \le (1 - u_w) V_w, \; w = \overline{1, W}, \quad (10)$$

- it is not allowed to keep particular files (or their fragments) in certain node

$$\alpha_w^z \sum_{f=1}^{Fz} x_{fw}^{z} \le 0, \quad z = \overline{1, Z}, \quad w = \overline{1, W}, \quad (11)$$

- it is an obligation to keep particular files (or their fragment) in certain nodes

$$\beta_{fw}^{z} \sum_{f=1}^{Fz} \left(1 - x_{fw}^{z} \right) \ge \beta_w^z, \; z = \overline{1, Z}, \; w = \overline{1, W}, \; f = \overline{1, F_z} \quad (12)$$

- each fragment must be placed at least in one node:

$$\sum_{w=1}^{w} x_{fw}^{z} \ge 1, \quad z = \overline{1, Z}, \quad f = \overline{1, F_z}, \quad (13)$$

Keeping in mind, that all criteria functions are to be minimalized, primary optimization problem, which takes all criteria functions in to consideration, have following form:

to find such $X^* \in X_{ogr}$

(set of conditions defined by formulas (9)-(13)), that :

$$F_i(X^*) = \min_{X \in X_{ogr}} F_i(X) \quad , i = \overline{1,4}, \quad (14)$$

3. Methods of Solving Defined Optimization Problems

Defined optimization problems can be easily turned into linear problems of binary mathematical programming. Due to large dimension of the problem (dimension of decision variable) in subsequent part of the paper global problem is divided into sequence of smaller problems.

Compression of problem is defined in virtue of file weights. File weight is considered as „importance" of file which is implicated by the access frequency to the file.

3.1. Divided problem of data files fragmentation, allocation and replication

Let g_z means weight of file z ($z = \overline{1, Z}$). Weight of a file is closely bound to semantics of examining computerized information system. System designer, who estimates it on the ground of given by user assumptions to the computerized information system, should give this weight. Weight can be implicated by access intensity to certain data file. However, one should be aware of cases in which it is impossible to estimate intensity. In those cases, weight is estimated taking in to account other characteristics.

Let k^i means order with number i, that is a vector of file numbers describing any order of the files. Then, it can be written:

$$k^i = (k_1^i, k_2^i, \ldots, k_z^i, \ldots, k_Z^i) \text{ for } i = \overline{1, z!} \quad (15)$$

where:

k_z^i - weight of file with number z ($z = \overline{1, Z}$) in order with number i.

Let k^0 means model order, it means order determining sequence of file numbers in descending order of file weights, where order is given by computerized information system designer (or user) and each file has assigned unique weight.

For that order, it can be written:

$$\underset{m,n=\overline{1,Z}}{\forall} \quad : \quad m < n \Rightarrow g_m^{k^0} \geq g_n^{k^0}, \tag{16}$$

Considered problem is divided into sequence of problems determined for each of data file.

For instance, as far as model order is concerned mentioned above problems have following form:

For each z ($z = \overline{1,Z}$) evaluate $X^{z^*} \in X_{ogr}^z$, that (17)

$$F_i^z(X^{z^*}) = \min_{X^z \in X_{ogr}^z} F_i^z(X^z), \quad i = \overline{1,4}$$

where:

$$F_1^z = \frac{1}{Z} \sum_{f=1}^{F_2} \left(\sum_{j=1}^{W} \frac{h_{f,j}^z}{\sum_{i=1}^{w} h_{f,i}^z} \sum_{w=1}^{W} x_{fw}^z d_{jw} \right), \tag{18}$$

$$F_2^z = \sum_{w=1}^{w} \sum_{f=1}^{f_z} x_{fw}^z \left(V_f^z + \Delta_f^z \right), \tag{19}$$

$$F_3^z = \frac{1}{U} \sum_{f=1}^{F_z} \left(\sum_{j=1}^{w} \frac{(h_{f,j}^z)^2}{\sum_{i=1}^{w} h_{f,i}^z} \sum_{w=1}^{w} x_{fw}^z d_{jw} \right), \tag{20}$$

$$F_4^z = \sum_{w=1}^{W} \left[1 - \prod_{f=1}^{F_z} \left(1 - x_{fw}^z \right) \right], \tag{21}$$

X_{ogr}^z - set of conditions for file with number z ($z = \overline{1,Z}$) defined, under assumption of ideal order k^0 in following manner:

$$X_{ogr}^z = \{ x^z \in [x_{fw}^z]_{F_z x W} : \text{fulfils conditions (22) – (26)} \}$$

$$x_{fw}^z \in \{0, 1\}, z = \overline{1,Z}, w = \overline{1,W}, f = \overline{1,F_z}, \tag{22}$$

$$\sum_{z=1}^{Z} \sum_{f=1}^{F_z} x_{fw}^z (V_f^z + \Delta_f^z) \leq (1 - u_w) V_w, w = \overline{1,W}, \tag{23}$$

$$\alpha_w^z \sum_{f=1}^{Fz} x_{fw}^z \leq 0, \quad z = \overline{1,Z}, \quad w = \overline{1,W}, \tag{24}$$

$$\beta_{fw}^z \sum_{f=1}^{Fz} \left(1 - x_{fw}^z\right) \leq \beta_w^z, z = \overline{1,Z}, w = \overline{1,W}, f = \overline{1,F_z} \tag{25}$$

$$\sum_{w=1}^{w} x_{fw}^z \geq 1, \quad z = \overline{1,Z}, \quad f = \overline{1,F_z}. \tag{26}$$

It is worth to point out, that it is necessary to solve z problems ($z = \overline{1,Z}$) quoted above, to evaluate solution of the primary problem (14).

3.2. Proposal of divided problems sequence solution

Similarly to primary problem, it is possible for divided problems (with four criteria functions) to utilize methods giving dominant or non-dominated solutions. In addition, in this case set of dominant solutions is empty. Additionally set of non-dominant solutions can be very numerous.

It is possible to solve the problem (14) using methods of Lagrange's factors. The method uses factors, which determines weights of particular criteria functions. In this case optimization problem has, determined for each $z = \overline{1,Z}$, form:

To find such $X^{z^*} \in X_{ogr}^z$ that: (27)

$$\overset{\circ}{F}{}^z(X^{z^*}) = \min_{X^z \in X_{og}^z} \overset{\circ}{F}{}^z(X^z) \tag{28}$$

where:

$$\overset{\circ}{F}{}^z(x) = \sum_{i=1}^{4} \alpha_i F_i^z(x) \tag{29}$$

where:

$\alpha_1, \alpha_2, \alpha_3, \alpha_4$ - Lagrange's factors fulfil following conditions:

$$\alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_3 \geq 0, \alpha_4 \geq 0 \quad , \quad \sum_{i=1}^{4} \alpha_i = 1 .$$

X_{ogr}^{z} - set of conditions for z'th file ($z = \overline{1, Z}$) defined above (22)-(26).

Problem in this form is a binary, one criteria linear mathematical programming problem. There are many methods of working out this king of problems, widely described in publications. This problem always gives non-dominant solution of problem (14).

Because of mentioned above activity, numerous sub-optimize solutions of problem (14) are received. Assuming, that designer chose α sets of LaGrange's factors with k distributed database file orders, he would receive $J = \alpha k$ solutions, that is:

$$X^* = \{ \quad X^{1*}, X^{2u}, \ldots X^{j*}, \ldots, X^{4J} \} \qquad (30)$$

Received solutions would be subsequently examined from ability of the system to survive. Those solutions are sub-optimized from the time of system response, communications load, file fragmentation and memory occupancy point of view. Solutions Selection from system ability to survive point of view, should at the end give sub-optimized solution of problem (14), which is most tolerant to destruction from tentative assumptions point of view.

3.3. Multi-criteria analysis of solutions of divided problem sequence

As mentioned above, exist $J = \alpha k$ solutions of problem (14), where α is a number of chosen by designer or generated sets of LaGrange's factors, and k is a number of order manners, which are implicated by importance of the files (their weights).

To solve problem (14), it can be used idea of mid-course solution. It relies on usage of ideal point. Solution algorithm of considered problem relies on performing following steps:

- Find J solutions of problem (14),
- Find ideal point y^* ,
- Propose distance measure between received solutions and ideal point, (for example using norm with parameter **p**),
- Evaluate sub-optimized solution of problem (14).

Sample outcomes (for i.e. J = 8) can be shown in the following table:

		F_1	F_2	F_3	F_4	$\|\|x\|\|$
1	x^{1*}	3	5	44	4	21
2	x^{2*}	67	7	32	7	43
3	x^{3*}	5	3	76	11	53
4	x^{4*}	12	8	90	9	29
5	x^{5*}	78	8	35	7	61
6	x^{6*}	33	9	44	6	44
7	x^{7*}	17	9	51	7	55
8	x^{8*}	21	12	23	10	43
	x^{*}	3	3	23	4	

In the table solution x^* is evaluated in the following way:

$$x^* = (\min_{i=\overline{1,J}} F_1(x^{i*}), \min_{i=\overline{1,J}} F_2(x^{i*}), \min_{i=\overline{1,J}} F_3(x^{i*}), \min_{i=\overline{1,J}} F_4(x^{i*})), \qquad (31)$$

Depend on measure (for certain factor p), it is received solution, which minimize the biggest deviation from ideal point.

Formulated optimization problem can be solved by one of known methods of evaluating optimized or sub-optimized solutions of linear binary programming problems.

4. Simulation Method of Selecting Solutions Tolerant to Disturbance

Because of applying introduced methods to acquire solutions of formulated problems, numerous solutions are received, which fulfil assumptions. To say, which solution is better and which worse from the tolerance to disturbance point of view, simulation method is used.

Disturbance, which is considered by author, is related to disturbance in distributed databases in military appliances. First of all, it concerns intentional destruction of computer network components during military operations.

Solutions, which are received using analytical methods, are used to perform simulation. In simulation experiment, computer network node destruction model is implicated by features of military operation. It is assumed that nodes are destroyed with certain probability and nodes are not able to be repaired (model of unregenerate object). Access to distributed database files exists as long as a node exists, which contains requested files. Conditions of the end of simulation experiment can regard accessibility to all files, certain files or the end of military operations. The discrete-event simulator SRBD was build with MODSIM II packet with SIMOBJECT library and MSVC. MODSIM III is an object-oriented, simulation tool specifically designed for modelling large, complex systems. Unlike general purpose languages such as C++ and Java, MODSIM III –captures both concurrent and interacting behaviours of system components in simulation:

- provides built-in statistical modelling and statistics gathering functions,
- comes with integrated graphics and animation functions,
- includes invaluable development aids such as run-time checking of object accessing, array bounds and memory.

Discrete Event Simulation Models can be realised and executed in many various ways. Methods and techniques adopted by many different users are depended of their own tools, software environment and preferences. The best-known standards in this area are as follows:

- Serial algorithms – for simple problems that can be solve on a single computer.
- Parallel Discrete Event Simulation (PDES) – for large problems, uses multiprocessor systems with synchronic algorithms.
- Parallel And Distributed Simulation (PADS) – real world elements and relationships are mapped into software and hardware infrastructure.
- Distributed Interactive Simulation – for large problems, software applications usually use well-defined standards for instance DIS, HLA.

Because of the problem size and software packet features the serial algorithm was chosen and implemented in the SRBD application.

Main requirements for the simulator are as follows:

- possibility of simulation of various computer network configurations – changeable number of nodes and connection structure,
- possibility of determining:
  - "read/write" query frequency to the specific file fragments,
  - probability function of network structure damages,
- possibility of characteristics obtaining, for instance access to the specific file fragments,
- implementation of the shortest path "node-node" algorithm,
- implementation of node destruction process.

Idea of such simulator is described as follow:

Step 1.
Definition of computer network structure: nodes and connections between nodes.

Step 2.
Features are given for each network node (similarly for lines).

Step 3.
After simulation is finished (certain passage number), expected value and standard deviation of random variable, which describes time of file (or fragment) access lack, can be read.

Step 4.
Time value of destruction, length of read/write path can be read and charts of those characteristics can be viewed, for each node or communication line.

After the end of simulation, it is possible to answer, which solution (one or more) is acceptable under assumed requirements (i.e. time of first file access lack or time of specific file access lack).

5. Conclusions

Simulation methods and techniques help visualize, analyse and predict the performance of various systems without the cost and risk of disrupting existing operations, or implementing new systems. They have also great impact on design process support of distributed databases. Nowadays, those methods are the way to check correctness and usefulness of mentioned above projects and usually the only way. Need and necessity to apply simulation methods is commonly known and accepted. Those methods should be broadly applied in the armed forces, where there is no other choice of solving certain problems. Testing of a computer system in wartime is a perfect sample of such problem.

Despite mentioned above considerations are based on military appliances, can be generalized to other domains. It regards particularly systems, which have to deliver information to users uninterruptible (i.e. banking systems).

Discovered Rule Filtering Using Information Retrieval Technique

Yasuhiko Kitamura*, Keunsik Park**, Akira Iida*, and Shoji Tatsumi*
*Graduate School of Engineering, **Graduate School of Medicine
Osaka City University, JAPAN
kitamura@info.eng.osaka-cu.ac.jp

Abstract

A data mining system can semi-automatically discover knowledge by mining a large volume of data, but the discovered knowledge is not always novel and interesting to the user. We propose a discovered rule filtering method to filter rules discovered by a data mining system and to produce ones that are novel and interesting to the user by using information retrieval technique. In the method, we rank discovered rules according to the result of information retrieval from the Internet. In this paper, we show the steps of discovered rule filtering by using a concrete example of clinical data mining and MEDLINE document retrieval. Preliminary results show that this method has merits in not only filtering discovered rules but also providing a new viewpoint to the rules to give a chance to invoke a new data mining process.

1. Introduction

As information technology becomes indispensable for our daily life, a huge amount of information is proliferated in the world. The speed and amount of the proliferation has been further accelerated by the advent of Internet and the available information is almost flooding us. From such a huge amount of various and noisy information, we need new tools to discover useful information or knowledge that meets demands of individual user. Active mining is a new direction of data mining and aims at discovering valuable knowledge for users in an efficient way by integrating data mining, information retrieval, and user reaction techniques [1].

As an approach to active mining, we have interest in integrating data mining and information retrieval techniques [3]. By using a data mining system, we can semi-automatically discover a number of rules hidden in a set of data, but each of the discovered rules can be classified according to the following characteristics.

(1) Does the rule express an important fact or not?
(2) Does the rule express a novel fact or a known fact?
(3) Does the rule express a fact that is interesting to the user or not?

Of course, we would like to discover rules that are important, novel, and interesting to the user. Conventional data mining systems mainly try to deal with the characteristic (1); the importance of rules, for example, by using a statistics approach. Some systems rank rules by using the precision and recall value of each rule. However, it is not easy to deal with other characteristics; the novelty of rule and the significance to the user because the novelty may change as the time goes on and the significance depends on the user's preference or interest.

To deal with characteristics (2) and (3), we try to utilize information retrieval results from the Internet. On the Internet, a huge amount of information is stored and is updated frequently. By retrieving latest information from the Internet, we may check whether a discovered rule is novel or not. Moreover, by monitoring the user's behavior of retrieving her preferred information, we may learn her preference and interest and may utilize them to check whether a discovered rule is interesting to her.

In this paper, we discuss the discovered rule filtering technique based on information retrieval from the Internet. In Section 2, we discuss the steps of discovered rule filtering. We here show an example when we apply the technique to a data mining task from a clinical examination database about hepatitis. We show preliminary results in Section 3 and conclude this paper in Section 4.

2. Discovered Rule Filtering in Hepatitis Data Mining

As a target of data mining, we use a clinical examination database of hepatitis patients, which is offered by the Medical School of Chiba University, as a common database on which 10 research groups cooperatively work in our active mining project. Some groups have already discovered some sets of rules. For example, a group in Shizuoka University analyzed sequential trends between a set of blood test data (GPT), which represents a progress of hepatitis, and other test data and has already discovered a number of rules, as one of them is shown in Figure 1.

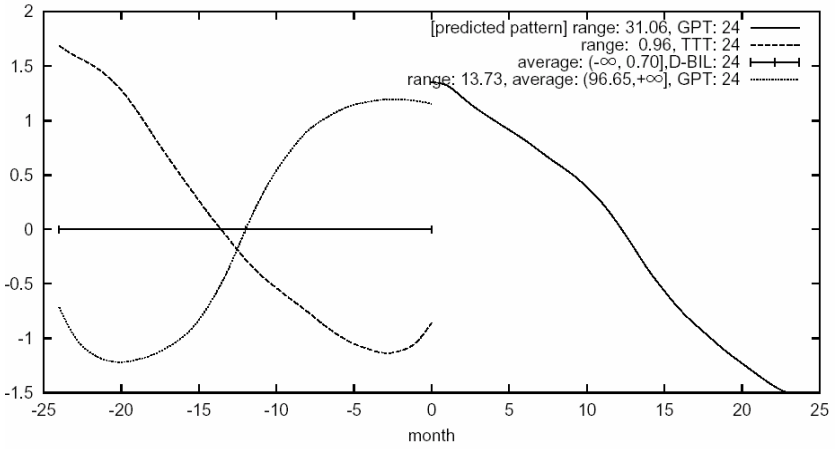


Figure 1. An example of discovered rule.

This rule shows a relation among GPT (Glutamat-Pyruvat-Transaminase), TTT (Thymol Turbidity Test), and D-BIL (Direct Bilirubin) and means "If, for 24 months, D-BIL stays unchanged, TTT decreases, and GPT increases, then GPT decreases for 24 months." A data mining system can semi-automatically discover a large number of rules by analyzing a set of data given by the user. On the other hand, discovered rules may include ones that are known and/or uninteresting to the user. Just showing all of the discovered rules to the user may not be a good idea and may result in putting a burden on her. We need to develop a method to filter the discovered rules into a small set of unknown and interesting rules to her. To this end, in this paper, we try to utilize information retrieval technique from the Internet.

When a set of discovered rules are given from a data mining system, a discovered rule filtering system first retrieves information related to the rules from the Internet and then filter the rules based on the result of information retrieval. In our project, we aim at discovering rules from a hepatitis database, but it is not easy to gather information related to hepatitis from the Internet by using naïve search engines because the Web information sources generally contain a huge amount of various and noisy information. We instead use the MEDLINE (MEDlars on LINE) database as the target of retrieving information, which is a bibliographical database (including abstracts) that covers more than 4000 medical and biological journals that have been published in about 70 countries. It has already stored more than 11 million documents since 1966. PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi) is a free MEDLINE search service on the Internet run by NCBI (National Center for Biotechnology Information). By using Pubmed, we can retrieve MEDLINE documents by submitting a set of keywords just like an ordinary search engine.

A discovered rule filtering process takes the following steps.

Step 1: Extracting keywords from a discovered rule

At first, we need to find a set of proper keywords to retrieve MEDLINE documents that relate to a discovered rule. Such keywords can be acquired from a discovered rule, the domain of data mining, and the interest of the user. These are summarized as follows.

- **Keywords related to attributes of a discovered rule.** These keywords represent attributes of a discovered rule. For example, keywords that can be acquired from a discovered rule shown in Figure 1 are GPT, TTT, and D-BIL because they are explicitly shown in the rule. When abbreviations are not acceptable for Pubmed, they need to be converted into normal names. For example, TTT and GPT should be converted into "thymol turbidity test" and "glutamic pyruvic transaminase" respectively.
- **Keywords related to a relation among attributes.** These keywords represent relations among attributes that constitute a discovered rule. It is difficult to acquire such keywords directly from the rule because, in many cases, they are not explicitly represented in the rule. They need to be included manually in advance. For example, in the hepatitis data mining, "periodicity" should be included when the periodicity of attribute value change is important.
- **Keywords related to the domain.** These keywords represent the purpose or the background of the data mining task. They should be included in advance as common keywords. For hepatitis data mining, "hepatitis" is the keyword.
- **Keywords related to the user's interest.** These keywords represent the user's interest in the data

mining task. They can be acquired directly by requesting the user to input the keywords or indirectly by using a relevance feedback technique as mentioned in Step 4.

Step 2: Gathering MEDLINE documents efficiently

We then perform a sequence of MEDLINE document retrievals. For each of discovered rules, we submit the keywords obtained in Step 1 to the Pubmed system [2]. However, redundant queries may be submitted when many of discovered rules are similar, in other words common attributes constitute many rules. The Pubmed is a popular system that is publicly available to a large number of researchers over the world, so it is required to reduce the load to the system. Actually, too many requests from a user lead to a temporal rejection of service to her. To reduce the number of submissions, we try to use a method that employs a graph representation, as shown in Figure 2, to store the history of document retrievals. By referring to the graph, we can gather documents in an efficient way by reducing the number of meaningless or redundant keyword submissions. The graph in Figure 2 shows pairs of submitted keywords and the number of hits. For example, this graph shows that a submission including keywords "hepatitis," "gpt," and "t-cho" returns nothing. It also shows that the combination of "hepatitis" and "total cholesterol" is better than the combination of "hepatitis" and "gpt" because the former is expected to have more returns than the latter.

Step 3: Filtering Discovered Rules

We filter discovered rules by using the result of MEDLINE document retrieval. More precisely, based on a result of document retrieval, we rank discovered rules. How to rank discovered rules by using the result of document retrievals is a core method of discovered rule filtering.

Basically the number of documents hit by a set of keywords shows the correlation of the keywords in the MEDLINE database, so we can assume that the more the number of hits is, the more the combination of attributes represented by the keywords is commonly known in the research field. We therefore use a heuristic such that "If the number of hits is small, the rule is novel."

The published month or year of document can be another hint to rank rules. If many documents related to a rule are published recently, the rule may contain a hot topic in the field.

Step 4: Estimating User's Preference

Retrieving documents by simply submitting keywords obtained in Step 1 may produce a wide variety of documents. They may relate to a discovered rule, but may not to the user's interest. To deal with this problem, we may request the user to input additional keywords that represent her interest, but this may put a burden to her. Relevance feedback is a technique that indirectly acquires the preference of the user. In this technique, the user just

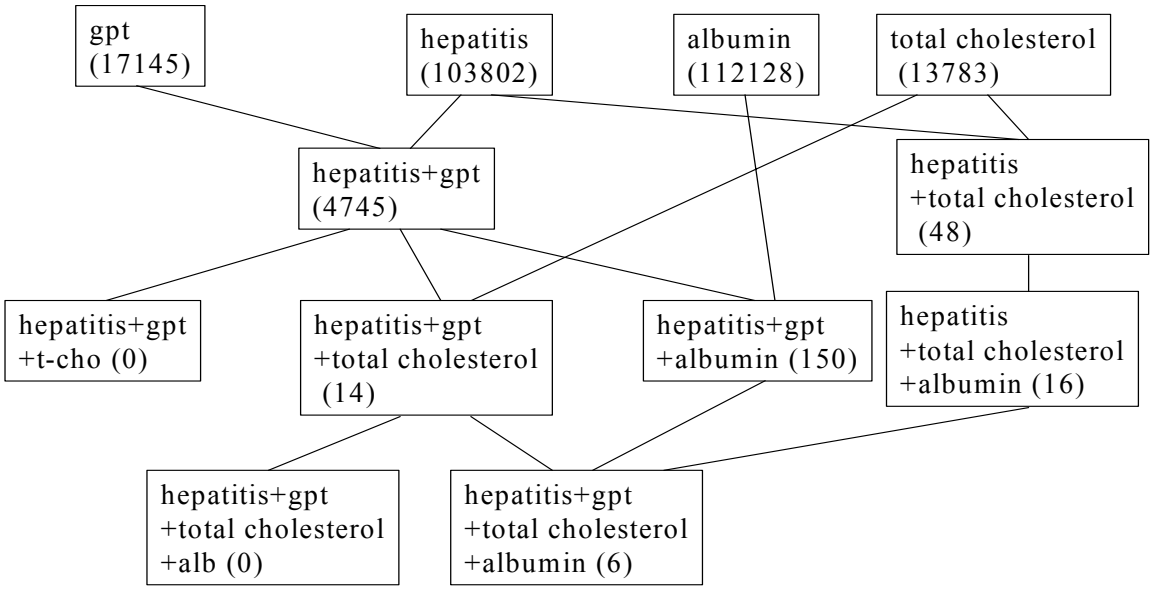


Figure 2. A graph that represents document retrieval history.

1: Hepatol Res 2001 Sep;21(1):67-75

Comparison of clinical laboratory liver tests between asymptomatic HBV and HCV carriers with persistently normal aminotransferase serum levels.

Murawaki Y, Ikuta Y, Koda M, Kawasaki H.

Second Department of Internal Medicine, Tottori University School of Medicine, 683-8504, Yonago, Japan

We examined the clinicopathological state in asymptomatic hepatitis C virus (HCV) carriers with persistently normal aminotransferase serum levels in comparison with asymptomatic hepatitis B virus (HBV) carriers. The findings showed that the thymol turbidity test (TTT) values and zinc sulfate turbidity test (ZTT) values were significantly higher in asymptomatic HCV carriers than in asymptomatic HBV carriers, whose values were within the normal limits. Multivariate analysis showed that the independent predictor of serum TTT and ZTT levels was the HCV infection. In clinical state, simple and cheap tests such as TTT and ZTT are useful for mass screening to detect HCV carriers in medical check-ups of healthy workers.

PMID: 11470629 [PubMed - as supplied by publisher]

Figure 3. A document retrieved.

feedbacks "Yes" or "No" to the system depending on whether she has interest in a document or not. The system uses the feedbacks as a clue to analyze the abstract of the document and to automatically find keywords that show the user's interest, and uses them for further document retrievals.

3. Preliminary Results and Discussion

To evaluate the feasibility of discovered rule filtering technique, we manually examined the number of retrieved documents for each of 30 rules discovered by Takahira Yamaguchi group at Shizuoka University. We used keywords only that are related to attributes of discovered rules and the domain. For 12 rules in 30, we could succeed to retrieve 7.3 documents in average. For 18 rules, we retrieved no documents. However, no hit does not always mean that the rule has a novel fact. Even when the rule contains no important facts, in other words it is just a garbage, it is likely that the system retrieves no documents. When the reliability of output from a data mining system is low, the discovered rule filtering does not work well.

We here discuss advantages of discovered rule filtering to deal with the characteristics (2) and (3) mentioned in Section 1. If we submit a set of proper keywords to the Pubmed system, we can roughly know how much work related to the keywords has been done in the research field. For example, if we submit "hepatitis GPT TTT," we have only 3 documents. On the other hand, if we submit "hepatitis GPT GOT," we have 1878 documents. The difference of the numbers is quite reasonable because GPT and GOT are well known blood test data to examine hepatitis. In addition, we know the relation between GPT and TTT has not been studied very much in the research field of hepatitis. Therefore, we may be able to conclude that a rule with attributes TTT and GPT looks more attractive than one with attributes GPT and GOT.

The number of retrieved documents changes depending on whether a user has a special interest. Let us assume a user has interest in the periodicity of attribute value. If we submit "hepatitis and GPT," we receive 4798 documents, but if we add "periodicity" to the keywords, we receive only 12 documents.

Of course, our information retrieval method based on keywords submission tends to produce noisy documents. We still need to improve the performance and we expect that the relevance feedback technique plays an important rule because it can narrow the space of document appropriately by using feedbacks from the user. We have not quantitatively examined how effectively the rule filtering technique works and left it as our future work.

However, we would like to report a side effect of showing discovered rules and related documents to a user (a medical doctor). In our preliminary experiment, at first we showed a discovered rule alone, shown in Figure 1, to the user and received the following comment (Comment 1). The discovered rule looks a part of common facts to the user.

Comment 1: "TTT shows an indicator of the activity of antic body. The more active the antic bodies are, the less active the hepatitis is and therefore the amount of GPT decreases. This rule can be interpreted by using well known facts."

We then retrieved related documents by using the rule filtering technique. The search result with keywords "hepatitis" and "TTT" was 11 documents. Among them, there was a document, shown in Figure 3, in which the user shows his interest as mentioned in a comment (Comment 2).

Comment 2: "This document discusses that we can compare type B virus with type C virus by measuring the TTT value of hepatitis virus carriers (who have not contracted hepatitis). It is a new paper published in 2001 that discusses a relation between TTT and hepatitis, but it reports only a small number of cases. The discovered rule suggests the same symptom appears not only in carriers but also in patients. This rule is important to support this paper from a standpoint of clinical data."

The effect shown in this preliminary examination is that the system can retrieve not only a new document related to a discovered rule but also a new viewpoint to the rule, and gives a chance to invoke a new mining process. In other words, if the rule alone is shown to the user, it is recognized just as a common fact, but if it is shown with a related document, it can motivate the user to analyze the amount of TTT depending on the type of hepatitis by using a large volume of hepatitis data. We hope this kind of effect can be found in many other cases.

4. Conclusions and Future Study

In this paper, we proposed the discovered rule filtering by integrating data mining and information retrieval techniques and showed the steps to develop a system. In a preliminary experiment, we show the technique contribute not only filtering discovered rules but also providing users a new viewpoint toward discovered rules and a motivation to invoke a new mining process. We believe this is a new approach to active data mining. Our future works are summarized as follows.

- **Evaluating the effect of discovered rule filtering.** We need to examine the relation between the novelty of discovered rule and the result of information retrieval.
- **Improving the performance of information retrieval.** By using the relevance feedback and other techniques, we need to improve the performance of information retrieval to meet the user's interest.
- **Developing a discovered rule filtering system.** We need to develop a system that automatically performs the process of discovered rule filtering.
- **Applying the discovered rule filtering technique to real-world research domains.** We are going to apply our system to support task of mining hepatitis data and show the effectiveness of the system.

Acknowledgement

We would like express out thanks to Professor Takahira Yamaguchi for giving us rules discovered at his laboratory for our preliminary experiment and also helpful comments for our work. This work is partly supported by Grant-in-Aide for Scientific Research (13131209) from the Ministry of Education, Culture, Sports, Science and Technology.

References

[1] Motoda, H. (Ed.), Active Mining: New Directions of Data Mining, IOS Press, Amsterdam, 2002.

[2] Kitamura, K., Nozaki, T., and Tatsumi, S. "A Script-Based WWW Information Integration Support System and Its Application to Genome Databases," Systems and Computers in Japan, Vol.29, No.14, 1998, pp.32-40

[3] Baeza-Yates, R. and Ribeiro-Neto, B. Modern Information Retrieval, Addison Wesley, 1999

Defect Classification Based on Association and Clustering

Iivari Kunttu[1], Leena Lepistö[1], Juhani Rauhamaa[2], and Ari Visa[1]

[1]Tampere University of Technology
Institute of Signal Processing
P. O. Box 553, FIN-33101 Tampere, Finland

[2]ABB Oy
Paper, Printing, Metals & Minerals
Automation
P. O. Box 94, FIN-00381 Helsinki, Finland

Iivari.Kunttu@tut.fi

Abstract

Image database mining is nowadays subject of great interest. Clustering of the images in the database is one basic task in the database mining. In this paper an image clustering procedure is introduced for an industrial imaging application. Database is indexed by extracting certain distinguishing features from the images. Clustering is made based on these features. The obtained cluster structures are associated with the real defects of the industrial process. The results of our experiments show that the clusters agree well with the traditional classification of the defects.

Keywords: image clustering, paper defects, image segmentation

1 Introduction

The growth of digital imaging during the last few years has affected many fields of human life. Nowadays digital imaging is popularly used in many industrial solutions concerning e.g. quality control and process control. In the process industry several on-line measurement systems are based on the digital imaging. As a result of this development, the amount of image data has increased rapidly. Consequently the sizes of different kinds of image databases in the industry have increased significantly. Therefore managing and mining of these databases has become necessary.

The goal in the industrial imaging applications in many cases is to divide different images into different classes. Usually the number of the classes is unknown, and therefore a clustering system is needed. The clustering procedure is based on certain features that describe the image content. These features can be for example color, shape or texture of the images. All of these features have been researched a lot during last years in the field of content-based image retrieval [12]. The color and shape features are discussed in [2], [8], and [13]. In [11] image clustering problem is discussed, and [9] provides an example of industrial solution to the image clustering.

In our research we approach this problem using paper industry as an example. In the paper mills, the avoiding and managing paper surface defects are key elements in quality control. The defects have also to be detected in order to prevent costly production disturbances during the further processing of the paper. Modern paper inspection systems are not only capable of detecting various defects but they can also produce gray scale images of the defects. This makes it possible to apply image analysis for defect identification.

Automatic classification of paper surface defects is quite demanding task. The defects are not always clear and their shape and gray level may vary also within the same class. Therefore, dividing the defects into classes is a difficult task even to an experienced operator. The normal background of the defect image is paper surface, with varying gray scale values. There has been made research work in the area of the surface defect inspection in the last decade. Image acquisition is presented in [6]. Detection [6] and segmentation of paper defects [7] were based on Self-Organizing Map (SOM).

In this paper we study clustering of paper defect images. In section two we present image preprocessing and feature selection. The features are used in clustering in section three. In the same section we make also an association between the selected defect clusters and known defect classes in the paper mill. The results of this paper are discussed in section four.

2 Image database indexing

2.1 Image preprocessing

As in data mining tasks in general, also in case of image databases we have to make some preprocessing before the features can be selected and extracted for database indexing. In case of paper defect images, the defect has to be extracted from its background. The background is paper surface, which forms a relatively constant distribution. In order to make this extraction, we have to use some image segmentation method [3],[11].

We have developed a specific segmentation procedure for defect images. It is based on the idea of Histogram Backprojection algorithm, of Swain and Ballard [13]. In histogram backprojection the idea is to use color histograms to locate certain objects in an image. In our approach, we use the gray level histograms of the defect image $\mathbf{I_d}$ and background image $\mathbf{I_b}$ to extract the defect from its background. The segmentation procedure is the following:

1. The numbers of gray levels of $\mathbf{I_d}$ and $\mathbf{I_b}$ are decreased from 256 (to 64 or 32, for instance). In this way we obtain quantized images $\mathbf{I_{dq}}$ and $\mathbf{I_{bq}}$.
2. The set of the gray levels belonging to the image background is defined as $\mathbf{G_b}$.
3. The gray levels of $\mathbf{G_b}$ are changed to zeros in the defect image $\mathbf{I_{dq}}$. The other levels of $\mathbf{I_{dq}}$ are changed to ones. In this way we obtain a binary segmentation mask **M**.
4. When we multiply bitwise the original defect image $\mathbf{I_d}$ by the mask **M**, the original defect is extracted from its background. In the resulting image the non-zero area is called $\mathbf{I_{seg}}$.

The procedure is presented in figure 1.

a)

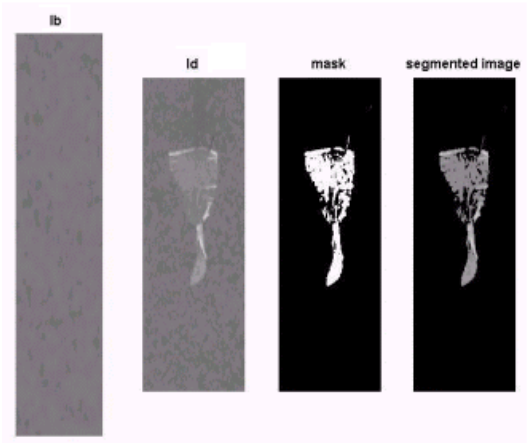


Figure 1. a) Background $\mathbf{I_b}$, defect image $\mathbf{I_d}$, mask **M**, and segmented image.

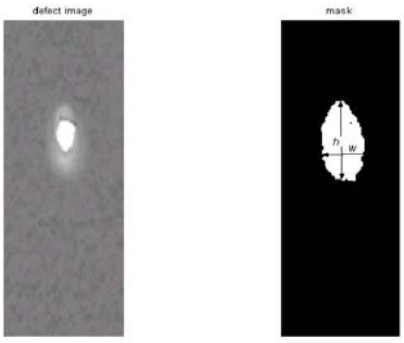


Figure 2. Shape measures *h* and *w*.

2.2 Feature selection

The features which are commonly used in image classification or retrieval [2], [12] are color, shape and texture. In case of the paper defect images we concentrate on gray scale and shape information of the defect.

The gray scale information of the defect can be extracted in many ways. One typical method is to use histograms [10],[13]. The histograms would be useful also in case of the paper defects, but in clustering process single numeric feature values are preferred to distributions. Therefore we describe this distribution of the defect by calculating mean of the gray levels (MGL) within the defect:

$$MGL = mean(\mathbf{I_{seg}}) \tag{1}$$

However, in some defects there are some areas, in which the gray levels vary significantly. In the same defect there may be white areas and black dots. In these cases mean gray level does not characterize the gray level distribution effectively. Therefore we use Gray Level Variance (GLV) as an additional feature.

$$GLV = \sum_{i=1}^{n} \left(\mathbf{I_{seg}}(i) - MGL\right)^2 / n \tag{2}$$

in which *n* is the number of the pixels in the defect.

The size information of the defect can be obtained from the segmentation mask. One effective way to describe the defect is to compare the area of the defect (A_{defect}) to the area of the whole image (A_{image}). Using this principle we obtain the feature Defect Area Ratio (DAR).

$$DAR = \frac{A_{defect}}{A_{image}} \tag{3}$$

Defect shape is an important feature in defect description. The shapes of the defects vary very much, since among the defects there are holes and spots, which have quite circular shape. In addition to them, long streaks and wrinkles are also typical. One approach to shape estimation is presented in figure 2. In this approach, the defect's maximum height (*h*) is compared to its maximum width (*w*). The resulting feature is Defect Shape Ratio (DSR):

$$DSR = \frac{h}{w} \tag{4}$$

3 Clustering

Han and Kamber [4] define the clustering as a process of grouping the data into classes or clusters. The objects in the cluster have high similarity in comparison to each other, but they are dissimilar to the objects in other clusters. Many methods and algorithms for clustering have been developed and the most popular of them are presented in [1], [4], and [5].

The features introduced in section 2.2 were used in our clustering experiments. The goal of the clustering

experiments was to clarify, how well the presented features were able to form paper defect clusters. In the association part the quality of the clustering result is measured by considering, how well the clusters are related to real defect classes.

Selection of the clustering method is an essential point in the process of clustering. In this case the most popular methods, *k*-means and *k*-medoids [1], [4] are not suitable, since they make only circular clusters in the feature space. As the figures 3 and 4 show, the cluster shapes are in many cases strongly elliptical. Therefore we developed an own clustering method for this purpose. Our method is able to make arbitrarily shaped clusters in the feature space. The idea of it is near *k*-nearest neighbor classification method [1], [5]. The clustering procedure is the following:

1. Some representative samples are selected among the data. This sample set is called training set
2. The samples that do not belong to the training set are clustered to the same cluster as majority of its *k*-nearest test set samples

The used distance metric in clustering is Euclidean distance [1].

3.1 Paper defect images

The variety of paper surface defects is quite large. Many defect types are common to all paper grades whereas some defects are specific to certain paper grades only. This is because different raw materials and equipment are used to manufacture different papers. For example, while making coated magazine paper, a coating layer is applied on the surface of the base paper. It is obvious that this process produces defects which are different from those emerging in making base paper. In addition, coating may even remove some minor defects of the base paper by covering them.

For this study we have selected 200 defect images appearing in base papers. The images have been taken of paper manufacturing process by a paper inspection system [6]. The objects in the images are typical paper surface defects. According to manual inspection the defects can be divided into nine classes. Among the data we selected three defect images to represent each defect class. These images were used as training data. Due to the number of the samples in training data set, the value of *k* was selected to be three in clustering.

3.2 Clustering of the defect images

We indexed the test set images by calculating features presented in section 2.2 for each image. After that several feature spaces were tested to obtain the best cluster structure. Clustering results of the defect images seemed to be the best when defect area ratio (DAR) and mean gray level were used. These features are presented in figure 3a.

The distribution of the data points and the training data are presented in figure 3a. The training data is presented in the same figure. The result of the clustering is presented figure 3b. In the manual inspection of the feature space, the clustering of the defects 1-6 seemed to be reasonable, because the obtained clusters contained clearly different defect types. On the other hand, the defects in each of these clusters were similar to each other. The defects in the clusters 7-9 represented several defect types, which were mixed together. To solve this problem, we applied a hierarchical clustering procedure, in which the defects of the clusters 7-9 were considered in other feature spaces. Because these clusters consisted of defects, whose shape and gray level varied significantly, we used defect shape ratio (DSR) and gray level variance (GLV) as clustering features. Figure 4a presents the defects in DSR-MGV-space. The training set samples are again used to make clustering in this feature space. The resulting new clusters 7-9 are presented in figure 4b.

3.3 Association

The images of the obtained clusters are presented in figure 5. In this part we are going to clarify the relation between the obtained clusters and the real defect classes. In other words, the goodness of the presented clustering method is measured by comparing the clusters to real defect classes.

The defects in cluster 1 seem to be similar. Actually the objects in this cluster are not real defects in paper. In fact, they are caused by loose paper flying beneath the cameras. The long shape of these objects are due to the imaging principle of line scan cameras and the difference between the speeds of flying paper and actual web. Defects of cluster 2 are also caused by the same reason, only the shape of the defect is narrower than the defects in cluster 1. In visual inspection the defects of cluster 3 form an obvious group. These defects are recognized as exceptional occurances due to sudden movements of paper web in cross direction. Some of the defects in this cluster are also edge defects. Clusters 4 and 5 consist of holes. Holes can be clean, such as in cluster 4, or caused by dirt, such as in cluster 5. Wrinkles are severe defects that often may cause the paper to break at later processing steps. Wrinkles can be narrow and faint (cluster 7) or wide with several folds and clearly visible (cluster 6). In addition to the wrinkles there are some other defects, like dirt, in cluster 6. The clusters 8 and 9 contain two types defects which cannot be totally separated from each other.

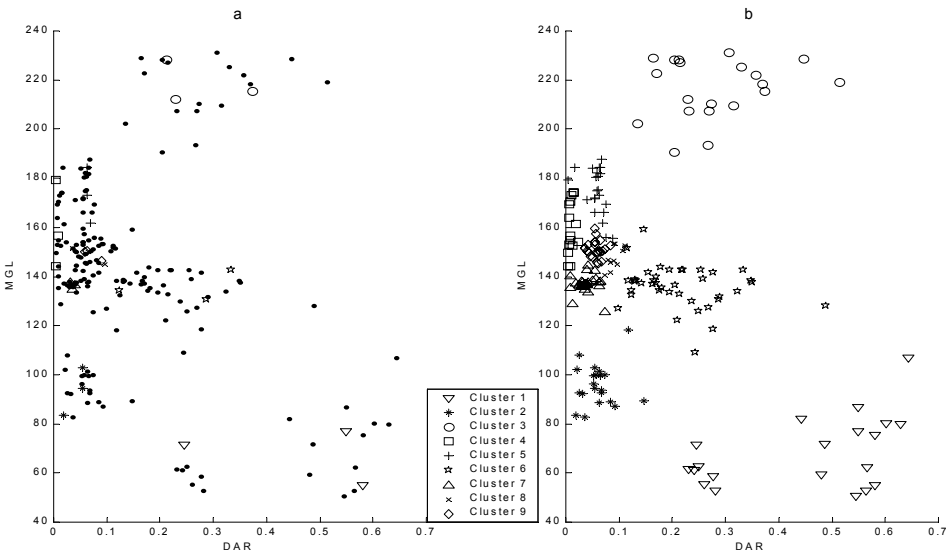


Figure 3. a) The images in terms of DAR and MGV. Training samples are marked in the figure. b) Result of the clustering.

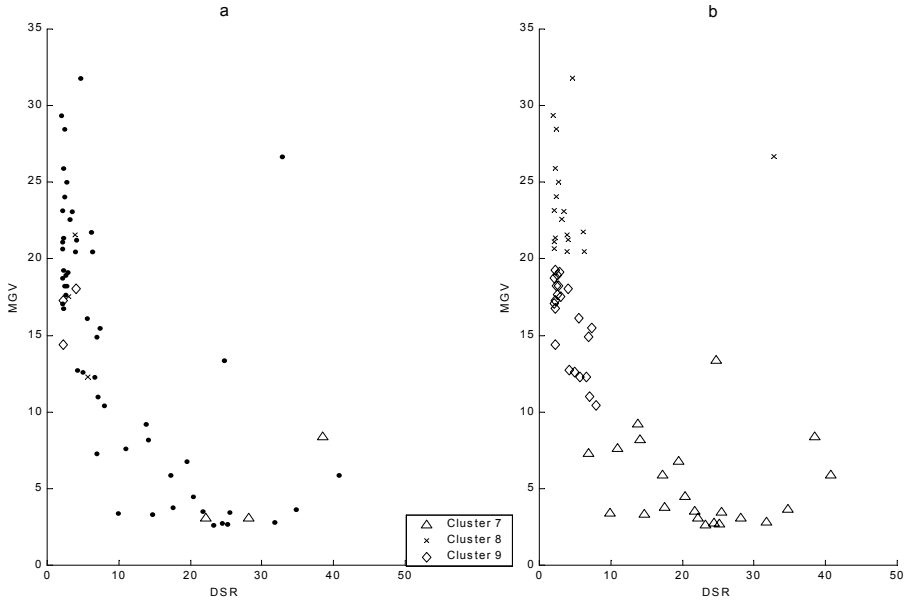


Figure 4. a) Samples of clusters 7-9 in terms of DSR and MGV. Training samples are marked in the figure. b) Result of the clustering.

The first of these defect types represents paper scraps calendered with the paper sheet. This type of defect causes translucent and weak areas in the paper. Another defect type in clusters 8 and 9 is dirt, which has in some images caused a hole to paper surface.

As a conclusion we can say that in clusters 1, 2, and 3 the defects have been correctly grouped based on their causes. Also defects in clusters 4, 5, 6, and 7, despite some exceptions, are distinguished based on the defect causes. Only the defects in clusters 8 and 9 had not been distinguished from each other. On the other hand, these clusters contain only two defect types. Consequently seven classes were classified correctly and remaining two classes had not been distinguished from each other. Table 1 presents the results of the clustering procedure.

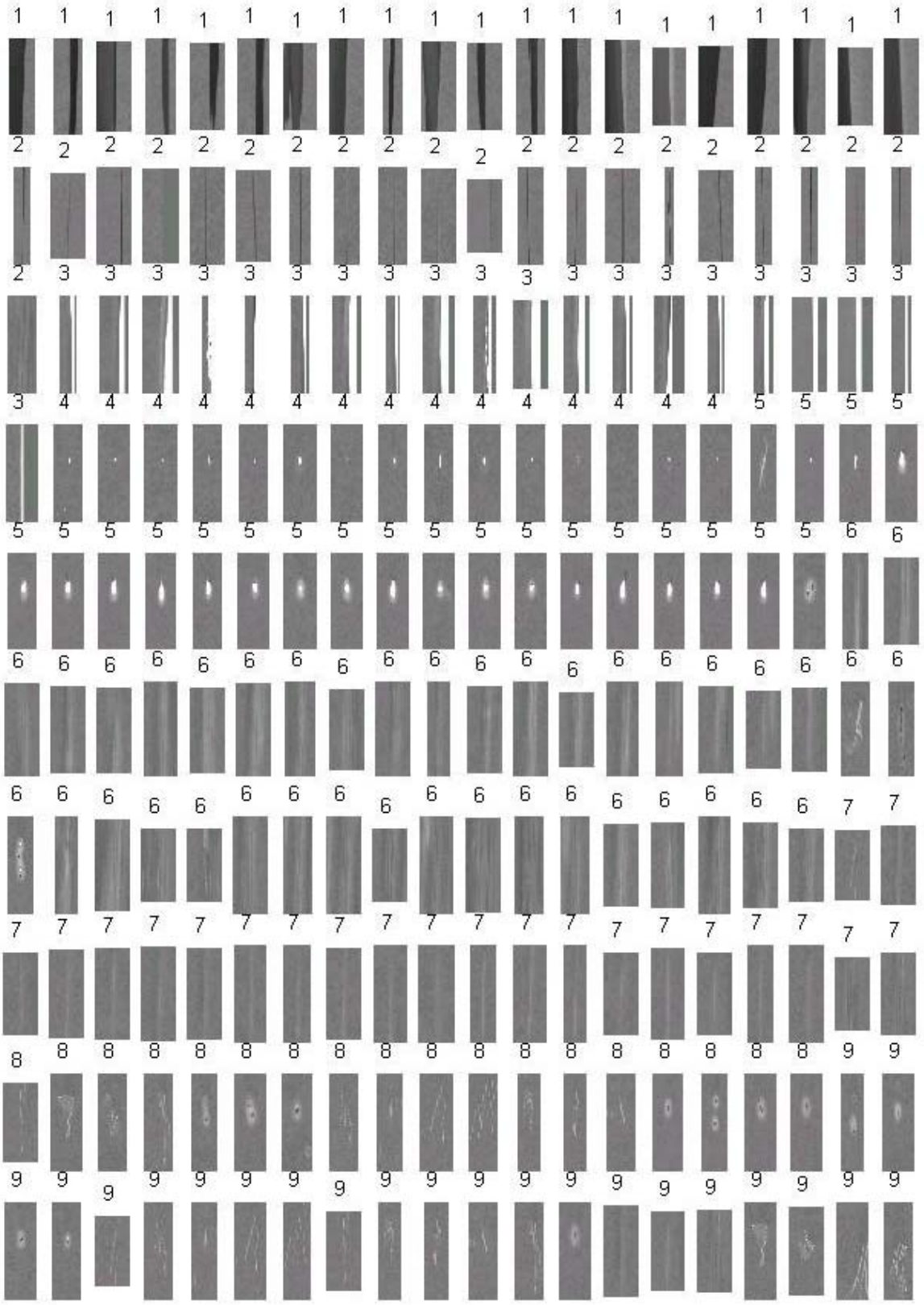


Figure 5. The images in the clusters.

Table 1. Results of the clustering.

Cluster	Description	% of correct defects in the cluster
1	Loose paper	100
2	Loose paper	95
3	Movement of paper web in cross direction	100
4	Clean hole	100
5	Hole caused by dirt	86
6	Wrinkle	92
7	Narrow wrinkle	100
8	Paper scraps calendered with the paper sheet	61
9	Dirt	23

4. Discussion

In this paper we approached a hierarchical solution for image clustering problem. This problem is common in many industrial imaging applications, for example in the quality control of the paper manufacturing process. In this work we used the paper surface defect images as testing database, but this clustering procedure can be applied also to other image types.

Our clustering procedure was not based on the commonly known clustering algorithms. Our algorithm was developed from *k*-nearest neighbor classification principle, and it made possible the arbitrarily shaped cluster structures. This algorithm was applied to defect images in hierarchical way. In this way it was possible to use several different feature spaces in the clustering procedure.

The feature selection has significant effect on the clustering result. The features presented in this work were based on the defect shape and gray level distribution. These features proved to be effective in the defect clustering. The majority of the defects (85%) were correctly clustered. In the association part a clear relation between the obtained clusters and the real defect causes was found. The association-based defect cause interpretation has a remarkable role in the defect analysis.

The limited size of the image database has certain effect on the clustering results. Even more accurate results could be achieved using a larger testing database. Also the proposed clustering algorithm should be tested using large database.

This work showed that it is possible to make an effective defect clustering using simple shape and gray level based features. Both the database indexing and the clustering are computationally very fast operations. Therefore they are very suitable for industrial imaging solutions, in which the databases are often large.

5. Acknowledgment

The authors wish to thank the Technology Development Centre of Finland (TEKES's grant 40397/01) for financial support.

References

[1] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, 2nd edition, John Wiley & Sons, 2001.

[2] T. Gevers and A. W. M. Smeulders. Pictoseek: Combining Color and Shape Invariant Features for Image Retrieval. *IEEE Transactions on image Processing.* vol 9, no 1, pp. 102-119, 2000.

[3] R. C. Gonzalez and R. E. Woods. Digital Image Processing. Addison Wesley, 1993.

[4] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Academic Press, 2001.

[5] D. Hand, H. Mannila, and P. Smyth, Principles of Data Mining, MIT Press, Massachusetts, 2001.

[6] J. Iivarinen and J. Rauhamaa, "Surface Inspection of Web Materials Using the Self-Organizing Map", *In D. P. Casasent (Ed.), Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision*, Proc. SPIE 3522, pp. 96-103, 1998.

[7] J. Iivarinen, J. Rauhamaa, and A. Visa, "Unsupervised segmentation of surface defects", In *Proceedings of 13th International Conference on Pattern Recognition*, Vol 4, pp. 356–360, Wien, Austria, Aug. 25–30, 1996.

[8] B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee. Shape Measures for Content Based Image Retrieval: A Comparison. *Information Processing Management,* vol. 33, no. 3, pp. 319-337, 1997.

[9] M. Niskanen, O Silvén, and H. Kauppinen. Color and Texture Based Wood Inspection with Non-supervised Clustering. *12th Scandinavian Conference on Image Analysis.* Bergen, Norway, June 2001.

[10] G. Pass and R. Zapith. Comparing Images using Joint Histograms. *Multimedia Systems,* vol. 7, pp. 234-240, 1999.

[11] E. J. Pauwels and G. Frederix. Nonparametric Clustering for Image Segmentation and Grouping. *Image Understanding,* vol. 75, no 1, pp. 73-85, 2000.

[12] Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., Jain, R. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 22, No 12, December 2000.

[13] M. J. Swain and D. H. Ballard. Indexing Via Color Histograms. *Proceedings of 3rd International Conference on Computer Vision*, pp. 390-393, 1990

Mining Knowledge from Hepatitis Data with Temporal Abstraction

TrongDung Nguyen, TuBao Ho, DucDung Nguyen, Saori Kawasaki
Japan Advanced Institute of Science and Technology
Tatsunokuchi, Ishikawa, 923-1292 Japan
nguyen@jaist.ac.jp, bao@jaist.ac.jp, dungduc@jaist.jp, skawasa@jaist.ac.jp

Abstract

The hepatitis database contains the results of laboratory examinations taken on the patients of hepatitis B and C during 1982-2001, and recently was given to challenge data mining research. This paper presents our approach to two problems of distinguishing hepatitis B and C, the relations between laboratory data and fibrosis stages, and effect of interferon treatment. The approach is based on temporal abstraction and the visual data mining system D2MS.

1. Introduction

This paper presents our approach to knowledge discovery in the hepatitis database. The approach is based on temporal abstraction and the visual data mining system D2MS (Data Mining with Model Selection) [3], [4]. In section 2, we briefly describe the mining problems, also the visual mining system D2MS and our framework for solving some posed problems. Section 3 describes the data preprocessing stage and the data table integrated for the purpose of mining temporal patterns that distinguish hepatitis B and C, the fibrosis stages, and the effect of interferon treatment. Section 4 presents procedures and results of basic temporal abstraction. Section 5 presents the procedure and results of complex temporal abstraction using visual system D2MS. Section 6 provides a discussion and conclusions. using visual system D2MS. Section 6 provides a discussion and conclusions.

2. Problems and our framework

Given the hepatitis database, the medical doctors posed the following problems:

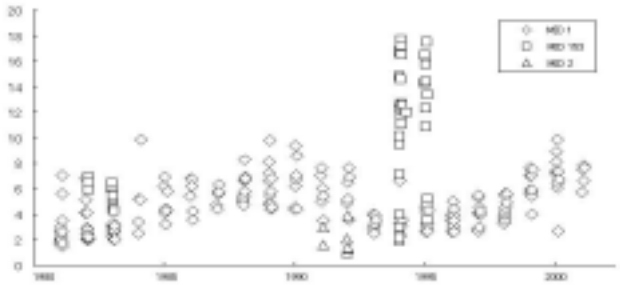


Figure 1: Different periods of examinations with irregular time-stamped points.

P1. Discover the differences in temporal patterns between hepatitis B and C.
P2. Evaluate whether laboratory examinations can be used to estimate the stage of liver fibrosis.
P3. Discover the relationships between the stage of liver fibrosis and the onset of hepatocarcinoma.
P3. Discover the relationships between hematological status and time to the onset of hepatocarcinoma.
P5. Evaluate whether the interferon therapy is effective or not.
P6. Validate if GOT and GPT can be used to measure the inflammation speed.

D2MS is a visual data mining system with visualization support for model selection [3], [4]. D2MS facilitates the trials of various alternatives of algorithm combinations and their settings. The performance metrics provided by the system is for a quantitative evaluation of the discovered patterns/models, while a qualitative evaluation can be obtained by effective visualization using D2MS's tools. The data mining methods in D2MS consists of CABRO to learned decision trees [9], CABRO-rule [8] and LUPC [5] to learn prediction rules. We propose a mining framework to deal with the hepatitis database and problems by using D2MS.

3. Preprocessing the hepatitis data

The preprocessing of the hepatitis database aims to extract from the hepatitis database a sub-dataset appropriate for mining purpose of each problem, the mining methods as well the available tools in D2MS. Our preprocessing of hepatitis data includes data cleaning, data integration, data reduction, and data transformation.

3.1. Preprocessing for problem P1 and P2

By combining the doctor guidance and the frequencies of attributes presented in [10], from 983 examinations, we selected 41 most significant examinations that can be divided into four groups:

(1) The most frequent examinations: GPT, GOT, LDH, ALP, TP, T-BIL, ALB, D-BIL, I-BIL, UA, UN, CRE, LAP, G-GTP, CHE, ZTT, TTT, T-CHO, oudan, nyuubi, youketsu;
(2) The high frequent examinations: NA, CL, K
(3) The frequent examinations: F-ALB, F-A2.GL, G.GL, F-A/G, F-B.GL, F-A1.G
(4) The less frequent but significant examinations: F-CHO, U-PH, U-GLU, U-RBC, U-PRO, U-BIL, U-SG, U-KET, TG, U-UBG, AMY, CRP.

These selected examinations will be used with other examinations in solving P2-P6. For the problem P2, we added other four examinations HBE-AB, HBE-AG, HBS-AB, HBS-AG according to the notices given by the doctors.

3.2. Preprocessing for problem P5

For problem P5, we have to firstly separate the patients into groups by response to IFN therapy based on the domain knowledge of doctors. There are four groups of patients who had been treated with IFN:

(1) Response: GPT data turned into normal range within 6 months after IFN therapy finished, and keep normal for more than 6 months.
(2) Partial response: GPT data turned into twice as high as normal range within 6 months after IFN therapy finished, and kept the level (twice as high as the normal rage) for more than 6 months.
(3) Aggravation: GPT data changed remarkably higher than the level before IFN therapy within 6 months after IFN therapy finished.
(4) No change: GPT data does not show change mentioned in (1)—(3).

Actually, these criteria are not concrete enough to group the data definitely, and can be used only as a general guide. To do that task of grouping we have developed a flexible *awk* program with several parameters that soften the above thresholds (these parameters will be refined with feedbacks from all successive steps of experiments).

4. Basic temporal abstraction methods

The fundamental problem here is how to transform temporal data of each patient into a record, i.e., how to transform multi time-stamped points of each patient in one examination into a fix number of values in the record. If transformed dataset can be obtained reasonably, many machine learning methods can be applied to it. Our framework to solve this problem is concerned with *temporal abstraction* (TA) methods.

The basic principle of TA is to move from a time-point to an interval-based representation of the data. The TA task can be defined as follows. The input includes a set of time-stamped data points (events) and abstraction goals. The output includes a set of interval-based, context-specific unified values or patterns (usually qualitative) at a higher level of abstraction.

Basic abstractions typically extract *states* (e.g., low, normal, high), and/or *trends* (e.g., increase, stable, decrease) from a uni-dimensional time-series. As the state and trend in each period are strongly related, we define a two-component structure of abstractions as *<episode, state & trend>*. The temporal abstractions in our work consist of the following tasks: (1) Determine context-sensitive episodes; (2) Determine abstracted states of episodes; (3) Trend movement analysis of episodes; (4) Discover rules by D2MS from abstracted values of episodes (complex TA).

Table 1: A procedure for determining context-sensitive episodes

Input: Integrated dataset with different data p using visual system D2MS. Section 6 provides a discussion and conclusions.oint intervals of examinations
Parameters: Expected length Δ of episode and the percentage ξ of patients having data within Δ
Output: Context-sensitive episode with length Δ for each examination.

Repeat for each examination:
1. Set Δ with the minimum value of the patient's data length (one unit is 1 month).
2. Compute the time-series length of each patient.
3. Compute the percentage ξ of patients having data within Δ.
4. Repeat $\Delta \leftarrow \Delta + \delta$ (by default $\delta = 1$) and compute the corresponding pairs (Δ, ξ) while Δ is still smaller than some given threshold.
5. Visualize pairs (Δ,ξ) in order to select the most

appropriate episode Δ.

4.1. Determination of episodes

This is certainly the most difficult TA task with hepatitis data in order to give significant episodes that characterize the patient's data according the mining purpose. The determination of episodes (b1) is usually dependent on and sensitive to the problem and context, while other tasks (b2) and (b3) are somehow more independent. Our solution is based on two assumptions, which are well accepted by the doctors, that almost patients has either hepatitis B or C that was determined when he/she started the treatment at the hospital, and the problem P5 has not been solved (people do not know the effect of interferon therapy). Therefore, for the problem P1 and P2, we decided to take a number of episodes for each examination so that (1) each episode has the same length for all patients, and (2) it is included in the patient period before the interferon treatment.

The remained task is how to determine an episode for each examination with an appropriate length Δ. The length Δ can be given by domain experts and depends on the processing purpose. In an arbitrary case, if we choose a large Δ many patients may not be taken into account (in other word, the percentage ξ of patients having data within Δ is reduced); and if we choose a small Δ many patients may be taken into account but we may risk to not use many important data points of these patients. Generally, there are two ways to identify a good trade-off pair (Δ, ξ): by expert opinion or by observation on real data. Figure 2 shows some statistics of the problem and observations of periods of 3, 6, 9, and 12 months. For example, 91% of patients have ALP examination data in a period longer than "3 months", but 74% of patients have ALP examination data in a period longer than "12 months".

Table 1 presents a procedure for determining context-sensitive episodes. The meaning of "context-sensitive" here is that a chosen episode for each examination depends strongly on data points measured by this examination. The procedure has been implemented as visual programs allowing the user to participate in deciding the most appropriate episodes. For the problem P1, based on this procedure, we have chosen two overlapping episodes, starting from the first day in hospital, of lengths 6 and 12 months for 33 examinations, and two overlapping episodes of lengths 3 and 6 months for examinations of U-BIL, U-GLU, U-KET, U-PH, U-PRO, U-RBC, U-SG, U-UBG. For the problem P2, we chosen the episodes on the same length but each of them ended at the day on which the biopsy stage is determined.

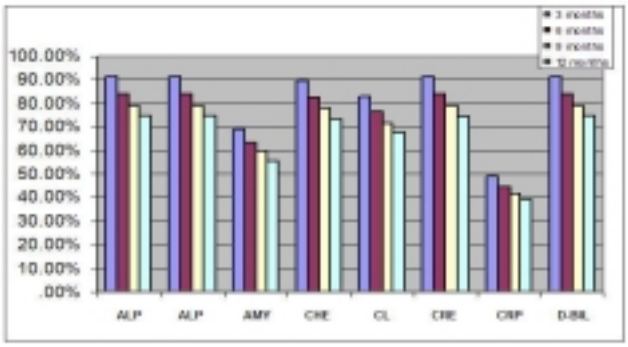


Figure 2: Distribution of patient's periods of examination data

4.2. Determine states of episodes

The patient's state of each examination on each episode can be characterized by the average of its values in the episode. This average value is then mapped to the label of one of sub-regions it belongs to. There are many possible choices of making sub-regions. Basically, we divide each examination range of states into:

- Seven sub-regions (states): "extremely high", "very high", "high", "normal", "low", "very low", "extremely low", or
- Five sub-regions (states): "very high", "high", "normal", "low", and "very low".

4.3. Trend movement analysis of episodes

The key idea is to use combinations of piecewise linear regressions on the episodes. As the episodes are chosen as significant period and they would not be split smaller. The piecewise linear regression allows us to approximate better the trend movements as done by using only linear regression [1]. Basically, we divide each examination range of trends into:

- Seven sub-regions (trends): "extremely fast increasing", "fast increasing", "increasing", "stable", "decreasing", "fast decreasing", "extremely fast decreasing", or
- Five sub-regions (trends): "fast increasing", "increasing", "stable", "decreasing", "fast decreasing".

By combining the state and trend on each episode, we transform all time-stamped data points of each patient on an examination into several qualitative values. Below is an example of some abstractions on ALB in the structure <*episode, state & trend*>, obtained for some patient when using two piecewise linear regression:

<ALB 6 months, very low & increasing-fast increasing>

5. Complex temporal abstraction using D2MS

This step aims to find significant temporal patterns that are combinations of basic abstractions obtained in the previous step. We describe in this section the experimental results of finding such relationships concerning the problem P1 and P2 when using the rule induction method LUPC of the visual data mining system D2MS.

LUPC is developed to learn prediction rules from supervised data. Each rule found by LUPC is a conjunction of attribute-value pairs that may present an interesting pattern. The main features of LUPC are (1) its ability of finding rules with associated domain knowledge (such as finding rules containing or not containing specified attribute-value pairs), as well finding rules for minority classes; (2) it is integrated with D2MS's rule visualizer and thus supports the user in selecting the appropriate rules resulted from different possible settings of parameters.

5.1. Complex temporal abstractions with high accuracy for problem P1

The performance of LUPC depends on several parameter specified by the user: α for min accuracy of rules, β for min coverage of rules, γ for maximal number of candidate rules in the beam search, and η for maximal number of attribute-value pairs to be consider. By varying these parameters we can find different sets of rules [5]. When using the setting with default parameters of α = 80%, β = 3, γ = 200, and η = 100, we found 119 rules characterizing the hepatitis B and 152 rules characterizing hepatitis C. Most of them cover from 5% to 15% of the whole patients and with accuracy (on training data) of at least 85%. Below are examples of these rules where the number following the examination name shows the length of associated episode, e.g., "CHE6" denotes the examination "CHE" with an episode of 6 months.

Rule 21: accuracy = 0.880 (44/50); coverage = 0.071 (50/702)
IF CHE6 = very low & decreasing-decreasing, AND
TP6 = normal & decreasing-decreasing
THEN class = hepatitis B

Rule 183: accuracy = 0.875 (35/40); coverage = 0.057 (40/702)
IF ALB6 = normal & decreasing-decreasing, AND
T-CHO3 = normal & decreasing-decreasing, AND
UN6 = normal & decreasing-decreasing, AND
ZTT6 = high & decreasing-decreasing
THEN class = hepatitis C

Such rules describe possible interesting relationships between basic temporal abstractions. Note that these relationships of laboratory temporal data reflect the domain knowledge specified by the doctor, such as the decreasing and increasing trends of certain examinations. An alternative is to find only rules with high accuracy. By setting α = 95% while the other parameters are kept without changing, we found 263 rules with accuracy 100% (on training data). Some of such rules have high coverage ratio. By using a 10-fold stratified cross evaluation with LUPC, we obtained an estimation of average error rate for these rules when diagnosing unknown patints as 17.820% ± 4.933%.

5.2. Complex temporal abstractions concerning common medical knowledge

The doctors group the changes of some significant examinations into two groups:

(1) Short term change: GOT (up), GPT (up), TTT (up), ZTT (up).
(2) Long term change: T-CHO (down), CHE (down), ALB (down), TP (down), PLT (down), WBC (down), HGB (down), T-BIL (up), D-BIL (up), I-BIL (up), ICG-15 (up).

One significant issue is to find complex temporal abstraction relating to such domain knowledge. We have done experiments for finding rules that contain examinations in these two groups, and detected subsets of discovered rules that are somehow different from common medical knowledge. The findings suggest that many patterns could be further considered, as they may be new, and will be shown in next subsections, to the common medical knowledge.

5.2.1. Complex temporal abstractions in the short term change group in problem P1.

Our experiments consist of running LUPC to find all possible rules containing examinations GOT, GPT, TTT, and ZTT with either increasing or decreasing trends. The outcomes are as follows:

- We found 147 rules (71 rules on hepatitis B and 76 rules on hepatitis C) with average accuracy 90.6% (on training data) that contain GOT, GPT, TTT, and ZTT with *increasing* trends, i.e., these rules are *consistent*

with the common medical knowledge. Most rules on hepatitis B concern with the increasing trends of TTT and/or ZTT, while most rules on hepatitis C concern with the increasing trends of GOT and/or GPT

- We found 222 rules (99 rules on hepatitis B and 123 rules on hepatitis C) with average accuracy 75.5% (on training data) that contain GOT, GPT, TTT, and ZTT with *decreasing* trends, i.e., these rules may be *not consistent* with the common medical knowledge. There is no dominant occurrence of decreasing trends of GOT, GPT, TTT, and ZTT in either hepatitis B or hepatitis C as in the previous case.

5.2.2. Complex temporal abstractions in the long term change group in problem P1.

The common medical knowledge given by physicians is "damaged liver cannot produce ALB any more" and "low T-CHO relates to damaged liver" (T-CHO (down), and ALB (down)). Our experiments consist of running LUPC to find all possible rules containing examinations ALB and T-CHO with either increasing or decreasing trends. The outcomes are as follows:

- We found 191 rules (82 rules on hepatitis B and 109 rules on hepatitis C) with average accuracy 82.3% (on training data) that contain ALB and T-CHO with *decreasing* trends, i.e., these rules are *consistent* with the common medical knowledge. There is no dominant occurrence of decreasing trends of ALB and T-CHO in either hepatitis B or hepatitis C as in the previous case.
- We found 274 rules (118 rules on hepatitis B and 156 rules on hepatitis C) with average accuracy 92.5% (on training data) that contain ALB and T-CHO with *increasing* trends, i.e., these rules may be *not consistent* with the common medical knowledge. The following two rules are examples of complex temporal abstractions that may suggest further investigations.

5.2.3. Discover relationships between stages of liver fibrosis and blood test data in problem P2.

The stages of liver fibrosis are determined by taking biopsy examinations. They reflex the progress of fibrosis and have five discrete values from F0 (mild) to F4 (severe). In the given data sets, the file that contains information about biopsy test is *bio_e.csv* (the number of records is 960). It has two most important attributes: "BIOPSY Exam_Date" gives the date when a patient took a biopsy test, and the result of the test "BIOPSY Fibrosis" stage. Our data abstraction solution for problem P2 is based on these two attributes.

According to the common knowledge of the doctors, there are several tests that concern with the product of liver. During the inflammation of liver their values change slowly because of the reserve capacity of live. These tests are T-CHO(down), CHE (down), ALB (down), TP (down), T-BIL (up), D-BIL (up), I-BIL (up), AMONIA (up), ICG-15 (up), PLT (down), WBC (down), HGB (down). In order to find *consistent* and *inconsistent* patterns with the common knowledge, we took advantage of the ability of D2MS to discover interesting rules by including these attributes values as core attribute-value pairs or excluding them form the attribute-values set. D2MS discovered totally 21 *consistent* rules and 5 *inconsistent* rules having minimum 80% of accuracy and cover at least 5 cases.

5.3. Complex temporal abstractions for problem P5

We ran LUPC with default parameters on the dataset mentioned in subsection 3.2 and got 44 rules including:

(1) 19 rules for *response* with sensitivity of 98.3% and positive predictive value of 76.3%
(2) 15 rules for *partial-response* (sensitivity: 71,4%; positive predictive value: 96,2%)
(3) 1 rule for *aggravation* (sensitivity: 40%, positive predictive value: 100%)
(4) 9 rules for *no-response* (sensitivity: 62.1%; positive predictive value: 94.7%)

We observed that, among short term change attributes, ZTT showed up in several rules, such as:

```
Rule 43: acc=0.857(12/14); cover=0.074(14/190)
IF      ALB6 = normal & decreasing-increasing AND
        ZTT3 = normal & decreasing-decreasing
THEN  class = response
```

Whereas, among long term change attributes, ALB has a very important role in problem P5 as it appeared in many rules, such as:

```
Rule 43: acc=0.842(16/19); cover=0.100(19/190)
IF      ALB3 = normal & decreasing-increasing AND
        T-CHO6 = normal & decreasing-increasing
THEN  class = response
```

6. Discussion and conclusions

We have presented a temporal abstraction approach to mining the time-series hepatitis data. This research relates to a very challenging and interesting domain of mining temporal data and trend detection. Though the project is on going, several lessons have been learned and in some issues could be further investigated.

(a) *Temporal abstraction* provides many advantages in mining temporal data, and typically suitable for many clinical tasks in medicine. Temporal abstraction, if it can yield meaningful abstractions, could allow us to apply symbolic learning methods to temporal data.

(b) The *hepatitis database* is a precious source for liver cancer research, but it also presents several interesting challenges to the data mining research. The most challenging feature is hepatitis clinic data were collected irregularly in regards to individual patients and examinations.

(c) The complex temporal abstraction done by data mining methods in D2MS allows us to discover combinations of basic temporal abstractions that characterize description patterns. The approach offers a descriptive way to distinguish temporal patterns of hepatitis B and C.

(d) The *interactive and visual system D2MS* provides us a powerful tool for complex temporal abstraction not only in combining obtained abstractions but also in visualizing them in order to give a better understanding of discovered relationships between basic temporal abstractions.

(e) The temporal abstraction approach presented in this paper is carried out in the scope of an on going project in collaboration with medical doctors. The initial results were obtained with their guidance and evaluation, in particular the background knowledge on hepatitis, the determination of episodes for the problem P1. The issues to be investigated in the next step include data preprocessing for other problems, the determination of unequal-length episodes and the trend detection on such episodes, the post-processing and interpretation of obtained complex temporal abstractions.

7. Acknowledgements

This research is supported by the project "Realization of Active Mining in the Era of Information Flood", Grant-in-aid for scientific research on priority areas (B). The authors would like to express their sincere thanks to doctors Katsuhiko Takabayashi and Hideto Yokoi of Chiba University Hospital for their guidance and collaboration.

8. References

[1] Bellazzi, R., Larizza, C., Magni, P., Monntani, S., Stefanelli, M., "Intelligent Analysis of Clinic Time Series: An Application in the Diabetes Mellitus Domain", *Artificial Intelligence in Medicine 20* (2000), 37-57.

[2] Han, J., Kamber, M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2000.

[3] Ho, T.B., Nguyen, T.D., Nguyen, D.D., Kawasaki, S., "Visualization Support for User-Centered Model Selection in Knowledge Discovery and Data Mining", *International Journal of Artificial Intelligence Tools*, Vol. 10 (2001), No. 4, 691-713.

[4] Ho T.B., Nguyen T.D., Nguyen D.D., Kawasaki S., "Visualization of Data and Knowledge in the Knowledge Discovery Process", *Active Mining: New Directions of Data Mining*, H. Motoda (Ed.) IOS Press, 2002, 229-238.

[5] Ho, T.B., Nguyen, D.D., Kawasaki, S., "Mining Prediction Rules from Minority Classes", *International Workshop Rule-Based Data Mining RBDM 2001*, Tokyo, October 20-22, 2001, 254-264.

[6] Horn, W., Miksch, S., Egghart, G., Popow, C., Paky, F., "Effective Data Validation of High-Frequency Data: Time-Point-, Time-Interval-, and Trend-Based Methods", *Computer in Biology and Medicine, Special Issue: Time-Oriented Systems in Medicine*, 27(5), 1997, 389-409.

[7] MedicineNet.com
http://www.focusoncancer.com/script/main/art.asp?articlekey=1917&rd=1

[8] Nguyen, D.T., Ho, T.B., Shimodaira, H., "A Scalable Algorithm for Rule Post-Pruning of Large Decision Trees", *5th Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD00*, April 2001, Hongkong. Lecture Notes in Artificial Intelligence 2035, Springer, 2001, 467-476.

[9] Nguyen, D.T., Ho, T.B., "An Interactive-Graphic System for Decision Tree Induction", *Journal of Japanese Society for Artificial Intelligence*, Vol. 14, N. 1, 1999, 131-138.

[10] Hatazawa, H., Sato, Y., Yamaguchi, T., "Rule Discovery Based on Sequential Pattern Analysis and Mining. In the Case Study of Chronic Hepatitis Datasets'', *JSAI SIG-KBS 56th Workshop*, Pusan, May 23-24, 2002, 55-60.

[11] PKDD02 challenge
http://www.cs.helsinki.fi/events/ecmlpkdd/challenge.html.

[12] Shahar, Y. and Musen, M.A., "Knowledge-Based Temporal Abstraction in Clinical Domains", *Artificial Intelligence in Medicine,* 8 (1996), 267-298.

[13] Shahar, Y., "A Framework for Knowledge-based Temporal Abstraction", *Artificial Intelligence,* 90 (1997), 79-133.

A Rule Discovery Support System for Sequential Medical Data — In the Case Study of a Chronic Hepatitis Dataset —

Miho Ohsaki[1] Yoshinori Sato[2] Hideto Yokoi[3] Takahira Yamaguchi[1]

[1]Faculty of Information, Shizuoka University
[2]Graduate School of Information, Shizuoka University
3-5-1 Johoku, Hamamatsu-shi, Shizuoka 432-8011, JAPAN

[3]Department of Medical Informatics, Chiba University Hospital
1-8-1 Inohana, Chuo-ku, Chiba-shi, Chiba 260-0856, JAPAN

Abstract

Although various rule discovery methods on sequential patterns has been proposed, there are few research to apply it to real, large-scale, and ill-defined datasets with many attributes and missing values. This paper discusses how pre-processing should be going and how a rule discovery support system should be developed, estimating the prognosis trend with an actual sequential medical dataset. We have done various important pre-processing for real medical data: unifying different names to the same entities, unifying different inspection cycle, discretizing time-series, and so on. Taking Das's framework, we then have obtained the rules, which come up with decision tree learning. Furthermore, we have visualized each rule on a graph consisted of diagnosis and prognosis sequential patterns. Medical experts have given us the following comment: some of the rules are interesting at a professional medical viewpoint and they may trigger a new discovery. Therefore, the case study has shown us that our system works well.

1 Introduction

Evidence-Based Medicine (EBM) is a medical practice method based on clinical evidence from systematic research. The concern with the relation between data mining and EBM has been growing for the last several years. However, many datasets obtained in daily clinics are sequential, large-scale, and ill-defined and need complex pre-processing for data mining. This problem is common for other real datasets that have similar properties.

However, many conventional studies on pre-processing focused on domain-independent methodology such as feature selection [5, 6]. There were few case studies that concretely explained domain-specific pre-processing for real data. Although there were some studies to apply data mining techniques to EBM, they were conducted by trial-and-error. It is then needed to establish the know-how and the methodology to apply data mining techniques to real medical datasets and to develop the support environment to discover medical knowledge for EBM.

Therefore, this paper targets a real chronic hepatitis dataset and aims to discuss the concrete methods of pre-processing, to develop a rule discovery support system, and to obtain practical and interesting rules for medical experts. In this paper, Section 2 notes the basic concept and the development of our rule discovery support system. Section 3 discusses the pre-processing method for a medical dataset. Section 4 shows the result of applying the system to the chronic hepatitis dataset. Finally, Section 5 concludes this paper and comments on the future work.

2 System Construction

2.1 Conventional Rule Discovery from Time Series

It is most important for EBM to objectively predict future symptom based on medical test results obtained for a certain term. Symptom changes at every moment, and diagnosis is conducted continually or intermittently. Many diagnosis data are numerical rather than symbolic. Therefore, the data mining technique to discover prediction rules from sequential and numerical data is suitable to a real medical dataset. Here, we discuss on the framework used in this research for rule discovery from time series.

There are two general methods to deal with numerical time sequences in machine learning. In one method, from a sequential data, we extract features such as frequency, distribution, and so on and regard them as attributes and a class. In the other method, we regard the symbol given to a typical pattern extracted from a sequence as attributes and a class. The former method has several problems: rule readability is low due to the indirect expression of a sequence, features depends on the kind of target datasets, and attribute dimension considerably increases. On the other hand, the latter method has many advantages: the rules with visualized patterns are easy to intuitively understand,

and we can control the abstraction degree depending on the number of patterns. We then adopted the latter pattern-based method.

Das et al. [1] proposed a pattern-based framework to discover rules from time-series and showed that their framework can discover quantitative and readable rules by applying it to actual datasets of marketing, telecommunication, and paleoecologic. This research applies it to an actual medical dataset.

Das's framework is shown in the left side of Figure 1. In this framework, subsequences are cut out from sequential data with a sliding window and representative patterns are extracted from the subsequences by clustering. Next, these patterns are regarded as attributes and classes, and the rules are discovered by a data mining scheme. Finally, the obtained rules are visualized as graph-based rules, namely the pattern combinations plotted on a graph.

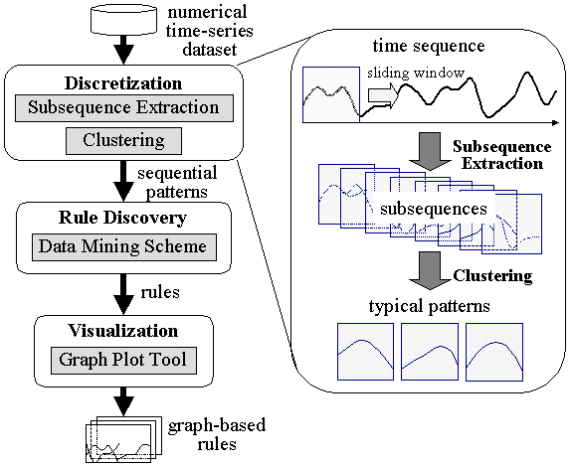


Figure 1: Framework to discover rules from time-series proposed by Das et al.

The details on the discretization process are shown in the right side of Figure 1. It consists of two phases: subsequence extraction and conversion into patterns by clustering. On the first phase, a subsequence $s' = (x_i, \ldots, x_{i+w-1})$ is cut out from a time sequence $s = (x_1, \ldots, x_n)$ by sliding a window of w-width at 1 sliding step.

A clustering method is used to form typical patterns on the second phase. We note on the method based on K-means algorithm [3] that is one of pattern extraction methods introduced by Das et al. [1]. This method initializes clusters by regarding randomly selected k subsequences as the centers of the clusters. For each cluster, it assigns the nearest subsequence to the center of the cluster to the cluster and regards the average of subsequences in the cluster as the new center of the cluster. This iterative process generates k clusters. The center of each cluster means a representative pattern of subsequences.

Once symbols are given to these patterns and are regarded as attributes and classes, various data mining schemes for symbolic data can be smoothly applied. Das et al. used association rule as a data mining scheme and obtained rules in a format such as "If A_1 and A_2 and ... A_h occur within V units of time, then B occurs within time T."

2.2 System Design and Development

We designed and developed a rule discovery support system for sequential medical data based on Das's framework. The construction of our system is shown in Figure 2. The system consists of two major components: pre-processing with two levels and domain knowledge feedback from an expert to the system.

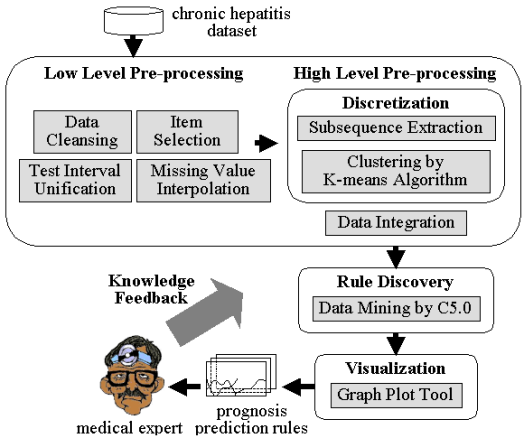


Figure 2: Proposed rule discovery support system for sequential medical data.

Generally speaking, a clinical dataset includes many kinds of medical tests, notation fluctuation, both of numerical and symbolic values, both of routine and thorough tests, various test cycles, and missing values. We then conducted the pre-processing at a low level that depends on medical domain knowledge and one at high level that does not depend as shown in Figure 2. We will explain the details on each processing part after the explanation of the target dataset in this research.

Instead of obtaining rules with high accuracy in a batch of data mining, the system aimed to polish up rules to make them practical and beneficial through iterative spiral processes. The discovered rules inspire an expert, and the expert returns his/her new knowledge to the system. Morik [7] advocated the importance of this approach and conducted some case stud-

ies based on it, and Motoda [8] advocated the expansion of this approach under the name of active mining. Our research is one of the case studies based on the active mining concept. Actually, we have developed and applied a rule discovery system using Das's framework to the same chronic hepatitis dataset in our previous research [11]. We get here from the research, using the comments on the previously obtained rules by a medical expert.

In the previous research, we used EM algorithm [2] as a clustering method and C4.5 [9] as a data mining scheme and obtained domain knowledge that GPT, one of diagnostic samplings, is the main measure to observe the condition of chronic hepatitis from a medical expert. We then obtained rules that predict the future trend of GPT using currently stocked data of various medical tests with our developed system. Figure 3 shows an example of the rules. The medical expert told us that this rule is interesting and commented on it as follows: It implies that GPT has a cyclic change and may become a new discovery, since the conventional common sense of medical experts was that GPT keeps a certain value in spite of slight change.

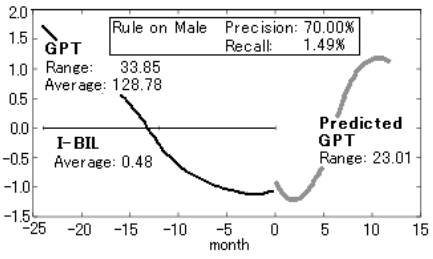


Figure 3: An example of rules obtained in our previous research on the same chronic hepatitis dataset as in this research.

This research aimed to discover rules to predict the future trend of GPT in the same way of the previous research. To confirm the hypothesis that GPT has a cyclic change, we extended an observation term and investigated GPT's long term trend. We used K-means algorithm [3] as a clustering method and C5.0, a commercial version of C4.5 [9], as a data mining scheme.

3 Pre-processing for Real Medical Data

3.1 Outline of Target Dataset

We used a dataset of medical test results of B and C hepatitis patients that was provided from Chiba university hospital. It was open as the common dataset of a data mining contest [10]. Although it includes the data on chronic and acute hepatitis, we focused on chronic hepatitis, since it is more necessary to grasp and predict the symptom of chronic hepatitis is rather than acute hepatitis.

The raw dataset consisted of five sets of medical test items: patient profile, diagnostic sampling meta-data, the results of diagnostic sampling, the results of liver biopsy, and the conditions of interferon medication. It includes 957 kinds of medical tests, 771 patients, and about 1,600,000 records. The details on each set of medical test items is as follows.

Patient Profile
Patient profile consists of identifier, sex, and birth date of a patient. Patient identifier was used to extract the subsequence of medical test results for each patient. Sex and age calculated using birth date were used as attributes to find out rules such as "If sex is female and age is greater than 50 and ..., then ..."

Diagnostic Sampling Meta-data
Diagnostic sampling is a generic term of blood and urine tests. Diagnostic sampling meta-data was not directly used in a data mining scheme, since it was the explanation on diagnostic sampling items.

Results of Diagnostic Sampling
Results of diagnostic sampling, the majority of data in our dataset, consist of many numerical values of test results and a few sentences to comment on them including patient identifier, the date of a test, measure unit, and so on. GPT, which was a class in our research, was one of diagnostic sampling. Patient identifier and the date of a test were used to extract the subsequence of a diagnostic sampling result for each patient. There are two kinds of diagnostic samplings: frequently conducted routine tests and rarely conducted thorough tests.

Results of Liver Biopsy
Liver biopsy is a test to examine liver damage using a small piece of liver tissue surgically took out. A medical expert judges the condition of liver by looking at the piece using a microscope and gives symbolic value such as 'fibrosis' as a liver biopsy result. The data on liver biopsy also included patient identifier, the date of a test, and virus type.

Liver biopsy has problems that it is rarely conducted due to heavy burden on a patient and that it is uneven due to subjective judgment. However, it has an advantage that it shows the progress of sickness more directly and clearly than routine tests do.

Conditions of Interferon Medication
Interferon is a specific medicine for hepatitis caused by virus. Conditions of Interferon medication consist of patient identifier, the date to start and to stop medication, and the number of medications. Although this conditions were essential to know the effect of medical treatment by medicine, the data on the internal

interferon quantity were not included in our dataset.

Here, we discuss on whether these sets of medical test items of hepatitis are common to that of other diseases. Generally speaking, patient profile, diagnostic sampling meta-data, and diagnostic sampling results are included in any sets of medical test items. Although liver biopsy itself is not conducted to inspect other diseases, similar tests that are rarely conducted but important exist for them. Interferon is a specific medicine for hepatitis and not common to the other disease. However, there are specific medicines for them. Therefore, the explanation on our sets of medical test items will be useful to deal with that of other diseases.

3.2 Low Level Pre-processing

We conducted data cleansing, item selection, test interval unification, and missing value interpolation as a low level pre-processing that depended on medical domain knowledge.

Data Cleansing
Before getting into the data cleansing to unify notation fluctuation, we removed diagnostic sampling meta-data, which only explain diagnostic sampling items, among five sets of medical test items mentioned in Subsection 3.1. In addition, we removed the data on acute hepatitis, because this research focused on chronic hepatitis.

The target dataset included noises such as different notation of the same medical test results, that of the same eras, and symbolic values that should be numerical values. Therefore, we modified and unified them based on domain knowledge on the name and the notation of medical test results that was given from technical books and experts on hepatitis. For example, symbolic values such as negative and positive or $-$ and $+$ were converted into numerical values, -1 and $+1$.

In fact, we conducted one of data cleansing, unification of notation fluctuation, after the discretization by pattern extraction mentioned in Subsection 3.3. The reason was why the pattern extraction automatically reduced the huge number of medical test items to be unified. This know-how will be practical for other medical datasets.

Item Selection
The number of medical test items, 930, was still huge even if the diagnostic sampling meta-data and the data of acute hepatitis were removed, and there were many items whose occurrence frequencies were too low to show their sequential trends. To avoid bad influence on the learning of a data mining scheme from redundant attributes, we removed results of liver biopsy and conditions of interferon medication as rare medical test items.

Test Interval Unification and Missing Value Interpolation
Here, we explain the both of test interval unification and missing value interpolation, because the interpolation was conducted for test interval unification. Generalization of a sequence needs instance sampling with a regular interval. We then investigated the proper sampling interval for all medical tests and conducted the merge and interpolation of data in a time-series to keep the regular interval.

Figure 4 shows the histogram of the number of medical test items for each sampling interval to find out a proper sampling interval. We adopted 28 days interval, since the number of medical test items was the largest in the sampling interval range from equal or more than 28 days to less than 56 days.

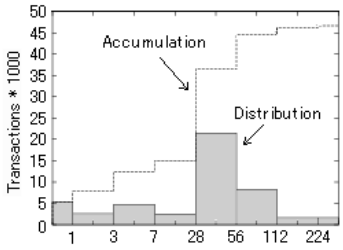


Figure 4: The histogram of the number of medical test items for each sampling interval.

In the case of the sampling interval less than 28 days, we merged all samples in the interval, namely all medical test results in 28 days were merged into one sample by averaging. In the case of the sampling interval more than 28 days, we filled in the blank with the average of two samples before and after the blank as a new interpolated sample.

The interpolation using long interval samples reduces data accuracy. Therefore, we did not interpolate a sample for the samples with an interval more than three months and divided the sequence into two sequences in such a case. We also removed medical test items that had the number of occurrence equal or less than 30 in five years.

3.3 High Level Pre-processing

We conducted discretization and data integration as a high level pre-processing that did not depend on medical domain knowledge.

Discretization
We cut out the subsequence of medical test results for each patient from the dataset using Das's frame-

work mentioned in Subsection 2.1. Five years term was adopted as the window size, since five years observation is needed to grasp the trend of chronic hepatitis at least. Consequently, this subsequence extraction removed the data of patients whose medical test terms were less than five years.

In addition, we removed the data of medical test items with less than two years term that were included in the subsequences. We then reduced the number of patients from 771 to 448, that of medical test items from 930 to 80, and that of records from 1,600,000 to 1,230,000.

Representative patterns were extracted from subsequences by clustering using K-means algorithm. To generate proper clusters, we randomly decided the number of clusters from two to eight. We then selected a set of clusters based on the two criteria: whether each cluster included at least ten instances and whether the distances among clusters were large enough. The former criterion was determined with the knowledge that the lower limit number of cases to confirm generality and reliability is ten at a viewpoint of medical science.

Data Integration

Finally, a new dataset consisted of five years subsequences were generated by putting together the discretized numerical sequential data and the symbolic data such as the name and the date of a medical test. We adopted the five years patterns of diagnostic sampling results as attributes: the patterns of blood test results and that of urine test results. We adopted the six months pattern of GPT that is the important measure of chronic hepatitis prognosis as a class.

4 A Case Study

We did a case study, applying the developed rule discovery support system (See Section 2) to the preprocessed dataset of chronic hepatitis (See Section 3). We then obtained 33 rules, selected 21 rules from them using the practical domain knowledge that the threshold of abnormal GPT value is over 100, and showed these rules to a medical expert as plotted graphs.

The expert gave us his comments on three rules; two rules were judged valuable and one rule was judged strange. In Figure 5, 6, and 7, the horizontal axis and the vertical axis mean month and the value of a medical test result, respectively. The graphs explain what GPT pattern will be brought out in the future six months by the patterns of medical test results in the past 60 months.

Rule 1: A rule judged valuable

Rule 1, which was thought highly by the expert, is shown in Figure 5. Refer the rule discovered in our previous research shown in Figure 3 before discussing on this rule. The previously obtained rule describes the change of GPT in the observation term from past 24 months to future 12 months. This rule inspired an expert to notice the possibility of GPT's cyclic change, and the expert said that GPT's cyclic change will be a significant discovery for the research on chronic hepatitis if it is proved.

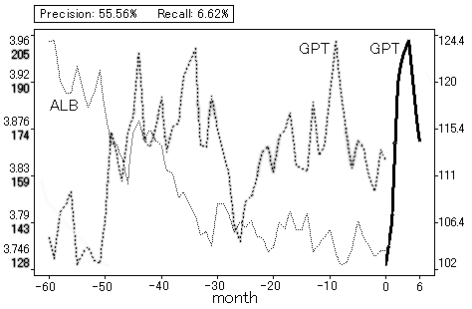


Figure 5: Rule 1, which was judged valuable by the expert.

Therefore, we tried to verify the hypothesis that GPT changes periodically and that its cycle is about three years by extending the observation term. The GPT observed from past 60 months to future 6 months in Figure 5 globally changes two times. Our hypothesis was more strongly supported by this result than the previous result did.

Rule 2: A rule judged valuable

Rule 2 shown in Figure 6 was also thought highly by the expert. It has the similar trend to that of Rule 1 and supports our hypothesis, namely the change of GPT with three years cycle.

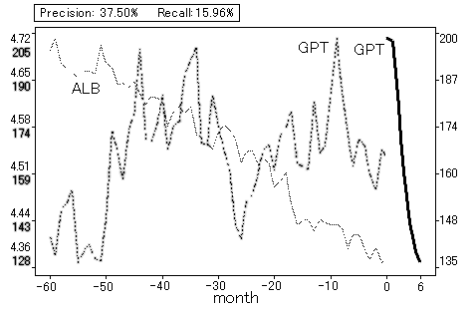


Figure 6: Rule 2, which was judged valuable by the expert.

Rule 3: A rule judged strange

Rule 3 shown in Figure 7 was strange for the expert. Although this rule describes that GOT increases after GPT reduced, this trend contradicts the domain knowledge on the relation between GPT and GOT. The expert explained that GPT and GOT must

change synchronically, since they are similar liver enzymes.

We then compared the raw data and the pattern for GPT and GOT and found that the difference between the raw data and the pattern for the both of GPT and GOT was too large in Rule 3. The patterns in Rule 3 were not representative and proper to reflect actual data. On the other hand, it was small enough in Rule 1 and Rule 2.

It is a severe problem for applying Das's framework [1] that clustering dose not work well and generates wrong patterns. Although we tried to generate proper clusters by controlling the number of patterns and that of instances in each pattern, that was not enough. We will keep discussing the clustering method for the pattern extraction.

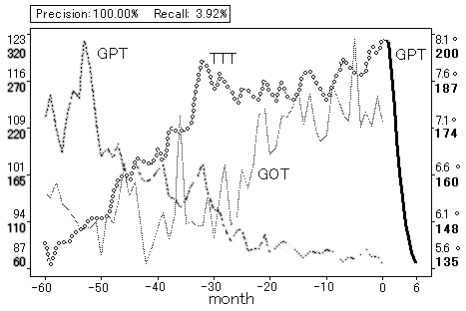


Figure 7: Rule 3, which was judged strange by the expert.

5 Conclusions

We targeted real medical data that are sequential, numerical, and ill-defined, discussed the pre-processing method, and developed a rule discovery support system for such data. We applied our system to a dataset consisted of medical test results for chronic hepatitis and obtained rules to predict prognosis.

The rules strengthened the reliability of the hypothesis that GPT changes with three years cycle formulated in our previous research. A medical expert thought the rules highly and commented that they are different from the common sense in medical science and have the possibility to be a new discovery. Therefore, we concluded that our pre-processing method, developed system, and knowledge feedback from an expert to the system were effective. Our methodology and know-how will be applied to other medical datasets.

We are going to continue this research to solve remained some issues. The diagnostic sampling data have a peculiar property that the occurrence frequency of a test item does not reflect the importance of the item. This representation bias may cause many valueless rules. We will discuss on how to reflect the item importance to the mining process. The other issues are as follows: the rule discovery between medicine and symptom and the improvement of the clustering method.

Acknowledgments

This research was supported by the Grant-in-Aid for Scientific Research on the Priority Area (B),13131205, by the Ministry of Education, Science, and Culture for Japan.

References

[1] Das, G., King-Ip, L., Heikki, M., Renganathan, G., and Smyth, P., "Rule Discovery from Time Series," Proc. of Int'l Conf. on Knowledge Discovery and Data Mining (KDD-98), New York, USA, pp.16–22 (Aug., 1998).

[2] Dempster, A. P., Laird, N. M., and Rubin, D. B., "Maximum likelihood from incomplete data via the EM algorithm," J. of the Royal Statistical Soc. Series B, vol.39, no.1, pp.1–38 (1997).

[3] Hartigan, J. A., "Clustering Algorithms," Wiley Publishers, New York, USA (1975).

[4] John, G. H., et al., "Irrelevant Features and the Subset Selection Problem," Proc. of Int'l Conf. on Machine Learning (ML94), New Brunswick, NJ, pp.121–129 (July, 1994).

[5] Kohavi, R. and John, G. H., "Wrappers for Feature Subset Selection," Artificial Intelligence, pp.273–324 (1997).

[6] Kononenko, I., "Estimating Attributes: Analysis and Extensions of RELIEF," Proc. of European Conf. on Machine Learning (ECML-94), pp.171–182 (April, 1994).

[7] Morik, K., Wrobel, S., Kietz, J. U., and Emde, W., "Knowledge Acquisition and Machine Learning," Academic Press (1993).

[8] Motoda, H. (eds.), "Active Mining," IOS Press (will be appeared in 2002).

[9] Quinlan, J. R., "C4.5: Programs for Machine Learning," Morgan Kaufmann Publishers, San Fransisco, USA (1993).

[10] "Hepatitis Dataset for Discovery Challenge," European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'02), Helsinki, Finland, http://lisp.vse.cz/challenge/ecmlpkdd2002/ (Aug., 2002) .

[11] Sato, Y., Hatazawa, H., Ohsaki, M., Yokoi H., and Ymaguchi T., "A Rule Discovery Support System in Chronic Hepatitis Datasets," Proc. of Int'l Conf. on Global Research and Education (Inter-Academia2002), Bratislava, Slovakia, pp.144–147 (Sept., 2002).

Active User's Response: Lessons from the Structure-Activity Relationship Analysis of Dopamine Antagonists

Takashi Okada
Center for Information & Media Studies, Kwansei Gakuin University
okada@kwansei.ac.jp

Abstract

Mining SAR (structure-activity relationship) is a complex problem that is typical of those encountered mining data. A recent analysis of dopamine antagonist activity produced valuable models of the interactions between antagonists and dopamine receptors. The cascade model was used together with methods recently developed for datascape surveys. Two analysts collaborated in interpreting the rules generated, one of whom has had a long career in drug design with a pharmaceutical company. This paper briefly introduces the problem, the data source, how attributes were generated, and the tasks involved in rule interpretation. Some important hypotheses are described. We discuss how they are drawn from the rule expressions, what kind of data stimulated the analysts, and the necessary improvements to the method to facilitate the active response of analysts.

1. Introduction

The importance of SAR (structure-activity relationship) studies relating chemical structures and biological activity is well established. Early studies used statistical techniques, and concentrated on establishing quantitative structure activity relationships involving compounds sharing a common skeleton. However, it is more natural to treat a variety of structures together, and to identify the characteristic substructures responsible for a given biological activity. Recent innovations in high throughput screening technology have produced vast amounts of SAR data, and the demand for a new data mining method to facilitate drug development has increased.

The author has already analyzed SARs with regard to the mutagenicity of nitroaromatic compounds [1] and the carcinogenicity of a variety of compounds studied by NTP [2]. The method used was the cascade model that we developed [3]. These studies demonstrated some of the characteristic features producing the activities in question. However, the method generated too many rules to allow reasonable interpretation of all of them.

Later, I pointed out the importance of the "datascape survey" in the mining process in order to obtain valuable knowledge. We added several functions to the mining software (DISCAS) to facilitate the datascape survey. The effectiveness of the method was demonstrated by applying it to a medical diagnosis problem [4, 5].

This new method was recently used to characterize the antagonist activity of dopamine receptors. Two people analyzed the resulting rules. Analyst A (the author) developed the system, and has a graduate-level knowledge of chemistry. Analyst B has had a long career in drug design as a chemist for a pharmaceutical company, but he has no experience in analyzing dopamine-related activity. The response of analyst B in the rule interpretation tasks was very active, leading to the success of mining. It is worth to analyze interactions between materials in the rules and the responses of the analyst.

The resulting interpretations of the rules were summarized as a draft paper for the recent Structure-Activity Relationship symposium [6]. The author asked two experts to comment on the draft: a professor of computer chemistry and a specialist in drug design working for a pharmaceutical company. The hypotheses presented in the draft were highly regarded by both these researchers.

This paper seeks to identify the factors affecting the response of analysts. Some functions of the DISCAS system play an important role in finding characteristic substructures, but several points remain to be improved to make it an analyst-friendly system. This paper traces the path between a rule expression and its subsequent interpretation. We point out the important functions that invoke active user responses. The next section briefly describes the mining process. Several interpretation tasks are explained in Section 3, where the lessons learned from these tasks are discussed.

2. Analysis of dopamine antagonist activity

2.1. Aims and data source

Dopamine is a neurotransmitter in the brain. Neural signals are transmitted via the interaction between dopamine and proteins known as dopamine receptors. There are five different receptor proteins, D1 – D5, each of which has a different biological function. Their amino acid sequences are known, but their three-dimensional structures are not yet established.

Certain chemicals act as antagonists for these receptors. An antagonist binds to a receptor, but does not function as a neurotransmitter. Therefore, it blocks the function of the dopamine molecule. Antagonists for these receptors might be used to treat schizophrenic patients. The structural characterization of these antagonists is an important problem in developing new schizophrenia drugs.

We used the MDDR database of MDL Inc. as the data source. It contains 1,364 records that describe dopamine (D1, D2, D3, and D4) antagonist activity. We used 1227 compound records as the learning set, after omitting 10% for use as a test set to check for misleading results caused by chance. There were 154, 383, 234, and 514 active compounds with D1 - D4 activity, respectively. Some of the compounds affected multiple receptors. The problem is to discover the structural characteristics responsible for each type of antagonist activity.

2.2. The cascade model and the datascape

The cascade model can be considered an extension of association rule mining. The method creates an itemset lattice in which an [attribute: value] pair is used as an item to constitute itemsets. Links in the lattice are selected and interpreted as rules. That is, we observe the distribution of the RHS (right hand side) attribute values along all links, and if a distinct change in the distribution appears along some link, then we focus on the two terminal nodes of the link. Consider that the itemset at the upper end of a link is [A {y}] and item [B {n}] is added along the link. If a marked activity change occurs along this link, we can write the rule:

IF [B {n}] added on [A {y}] Cases: 200 ==> 50
THEN Activity: .80 .20 ==> .30 .70 (y n) BSS = 12.5
THEN C: .50 .50 ==> .94 .06 (y n) BSS = 9.68
Ridge [A {n}]: .70 .30 / 100 ==> .70 .30 / 50 (y n)

where the added item [B {n}] is the main condition of the rule, and the items at the upper end of the link ([A {y}]) are considered preconditions. The main condition changes the ratio of the active compounds from 0.8 to 0.3, while the number of supporting instances decreases from 200 to 50. BSS means the between-groups sum of squares, which is derived from the decomposition of the sum of squares for a categorical variable. Its value can be used as a measure of the strength of a rule. The second "THEN" clause indicates that the distribution of the values of attribute C also changes sharply with the application of the main condition. This description is called the *collateral correlation*. Its BSS value is also shown, but it does not affect the selection of the rule.

The last line includes ridge information. This example describes [A {n}], the ridge region detected, and the change in the distribution of "Activity" in this region. Compared to the large change in the activity distribution for the instances with [A {y}], the distribution does not change on this ridge. This means that the BSS value decreases sharply if we expand the rule region to include this ridge region. This ridge information is expected to guide the survey of the datascape.

A rule candidate link found in the lattice is first greedily optimized in order to give the rule with the largest local BSS value, changing the main and preconditions. This process is useful for decreasing the number of resulting rules, since many rules converge on the same expression. The resulting rules are expressed after organizing them into principal and relative rules based on the overlap of supporting instances. This function is useful for decreasing the number of principal rules to be inspected, and to indicate the relationships among rules.

2.3. Attribute generation and selection

Two kinds of explanation attributes were generated from the structural formulae. The first group consists of four physicochemical estimates: the HOMO and LUMO energy levels, the dipole moment, and LogP. The other group is the presence/absence of various structural fragments. Obviously, the number of all possible fragments is too large. We generated linear fragments with lengths shorter than 11. One of the terminal atoms of a fragment was restricted to be a heteroatom or a carbon constituting a double or triple bond.

We considered linear fragments expressed using the following four schemes:

(1) Fragments are expressed by constituent elements and bond types, *e.g.*, C:C-C-N-C=O, where ":" is an aromatic bond.
(2) The number of coordinating atoms and the presence/absence of attached hydrogens are added to the terminal atom, *e.g.*, C3H:C-C-N-C=O1.

(3) Coordination numbers and hydrogens are added to the intermediary atoms in the fragment, *e.g.*, C3H:C3-C4H-N3H-C3=O1.
(4) If a fragment has a branch at an intermediary atom, the neighboring bond and the atom on the branch are also specified, *e.g.*, C3H:C3(:C3)-C4H-N3H-C3(-C4H)=O1.

Schemes (1) to (4) generated 6,622, 12,140, 13,972, and 16,247 attributes from the 1,227 compounds, respectively. The DISCAS system automatically selects those attributes with a balanced presence/absence distribution. The threshold value for selection is determined by a parameter that controls the level of details in the lattice expansion. When we set this parameter (*thres*) to 0.12, the above 4 schemes employed 65, 57, 50, and 32 attributes, respectively. This shows that the number of useful attributes for mining decreases when we use a detailed description. Conversely, a simpler expression causes ambiguity in the interpretation of rules since an analyst cannot recognize the detailed features of a fragment. Consequently, we used the third scheme for the real mining task. Setting the parameter to *thres*=0.10 resulted in 74 attributes for mining.

2.4. Calculation and interpretation of rules

For the calculations with the DISCAS ver.3 software the parameters were set at *minsup*=0.01, *thres*=0.1, *thr-BSS*=0.015, *min-rlv*=0.7. These parameters are defined elsewhere [3, 4, 5]. As an example, the D1 antagonist calculation required about 9 minutes to generate a lattice with 76,441 nodes, and 4.5 minutes were spent polishing the rule expressions for the datascape survey. Fifty-one candidate links for rules were optimized to give 12 rules with local maximum *BSS* values, and these were then organized into two principal rules. The first principal rule was attached to 4 *ULrelative* and 3 *Lrelative* rules, while the second principal rule had only 1 *ULrelative* rule. (These relative rules are defined elsewhere [5].)

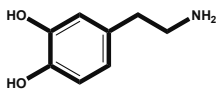
First, Analyst A read every principal rule and its associated collateral correlations and ridges. When the analyst found a characteristic feature, he visualized the distribution using pie or bar charts generated with Spotfire software and recorded them in a word processor. He also recorded the structural formulae of the compounds selected by the main condition of a rule. At this stage, Analyst A noticed that there was much material to arrange, but he did not develop any concrete ideas. Then, Analyst B reviewed the recorded visualizations and structural formulae, as well as the original rules. Although this process was intended to generate insights in the mind of analyst B, no valuable findings were made. Therefore, both analysts navigated all the rules again, visualizing the distributions of various attributes and the supporting structural formulae. After this process, we developed a reasonable set of hypotheses. Although the interpretation of the results is ongoing, it has already been given a good evaluation.

3. Lessons from the interpretation task

Aside from the participation of Analyst B*, the first reason for success in interpretation was a smaller number of rules*. The numbers of principal rules were limited to 2, 2, 4, and 11 for activities D1 to D4, respectively, which motivated the analysts to read and interpret all of the rules. In previous analyses, there were so many rules that we were forced to select some rules with large BSS values, using an arbitrary threshold value. In such a situation, the analysts need a strong incentive to inspect all the rules in detail. The interpretations discussed in the following paragraphs would have been impossible if there were 10 times as many rules.

The main condition of the first principal rule for D1 activity showed the presence of a C3H:C3-C4H-C4H-N3 fragment, while that of the second principal rule was the presence of C3-C4H-C4H-N3. The latter fragment is easily judged to be a substructure of the first, so both are thought to have the same meaning. In fact, collateral correlation of the second rule confirmed that the first fragment appears in 96% of the supporting compounds. In this case, *the collateral correlation information is important to ensure the hypothesis*.

After reading these rules, Analyst B immediately pointed out that the rule is reasonable, because this fragment is a substructure of dopamine (shown to the right) with substitutions at the nitrogen atom. Analyst A should also have reached this conclusion readily, if he had hypothesized on the interaction between the dopamine molecule and its receptor. The difference in the results with the two analysts showed *the importance of having a hypothesis in the process of inspecting rules.*



A precondition of the second rule was the absence of C3-C4H-C4H-C4H-N3, and its presence was part of its ridge information. Compared with the main condition, this fragment contains one more -C4H- between C3 and N3. The pie graphs in Figure 1 show the proportion of active compounds (dark area) using the pre- and main conditions as the x- and y- axes, respectively. We readily notice that the change is small (.13 => .15) in the ridge region to the right, while a sharp increase (.25 => .74) is observed in the rule region to the left.

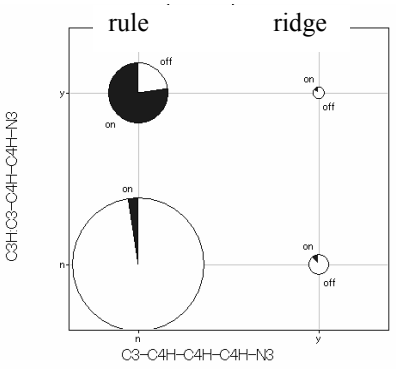


Figure 1. Change in the D1 activity ratio

Analyst B raised a question on the optimal length between the N and the phenyl group before inspecting the rules. Therefore, he concluded that the optimal length is two -C4H-'s or less after he saw this figure. *The usefulness of visualization using ridge information was indicated for formulating a detailed hypothesis.*

The first principal rule for D3 activity used the presence of O1 as the main condition, while one of its preconditions was the absence of C3=O1. As the most common fragment found in organic chemistry containing O1 is C3=O1, this rule initially seemed odd to both analysts. However, it was readily noted that most of the supporting compounds contained an $-S(=O)_2-$ fragment. *This demonstrates the necessity of the capability of instance inspection by browsing structural formulae.*

The selection of fragment attributes is also important. When common functional groups are included in the attributes, the above question does not arise because the $-S(=O)_2-$ fragment appears in the collateral correlation. Therefore, *analysts should use those attributes that can help in intuitive interpretation, even if their one-sided distribution prevents their appearance in derived rules.* These two lessons seem to be domain dependent, but they suggest that attributes should be discernable by analysts.

The analysis of D4 activity provides an interesting example of an interpretation task. The first principal rule used the presence of C3:N2 as the main condition, increasing the active compounds ratio from .43 in 787 compounds to .86 in 250 compounds. This suggests the relevance of a heteroaromatic ring containing a nitrogen atom, but a detailed hypothesis could not be constructed using this rule.

One of its relative rules raised the ratio of active compounds from .40 in 513 compounds to .94 in 81 compounds. This rule was classified as a relative rule, as 66 out of 81 compounds are shared with those of the principal rule after applying the main condition. The main condition was the presence of C3:C3:N3H and the preconditions were the absence of C3H:C3:N3H and the presence of N3H. These conditions are completely different from those of the principal rule. *A relative rule can give alternative explanations for a subgroup of compounds specified by the principal rule.*

This relative rule is interesting, as one of the collateral correlations showed that the D2 activity decreased from .36 to .02. Figure 2 shows the change in the active ratios for D4 (outside) and D2 (inside) after filtering according to the presence of N3H. Another interesting point is the similarity between the attributes for the x- and y-axes, which differ only in the attachment of hydrogen at the terminal aromatic carbon. The difference caused by the presence/absence of these two fragments should appear in the two pie graphs in the upper left and lower right. They show opposite D4 activity ratios, while almost no D2 activity is observed in both pie graphs. These speculation steps do not need the expert knowledge of a pharmacologist, but *suggest the necessity for careful inspection of a rule and its visualization*.

Browsing the structural formulae supporting these two

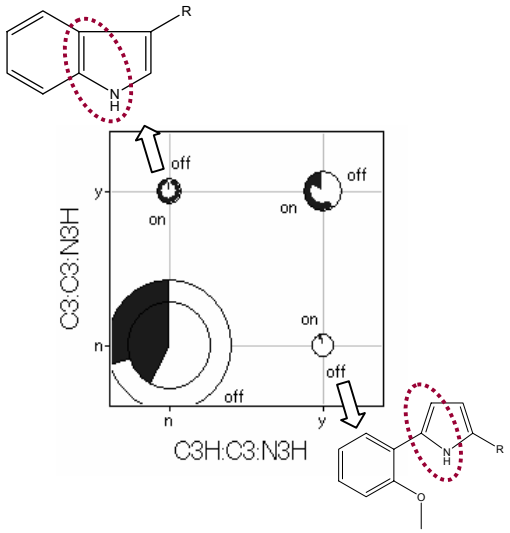


Figure 2. Changes of D4 (outside) and D2 (inside) activity ratios

pie graphs clarified the above difference. Typical supporting structures for these graphs are shown by the structural formulae in the figure, where the dotted circles indicate the fragments used as the axes. Both analysts noticed a clear difference between the two structures: the presence of a hydrogen bond between the NH and O in the lower right structure.

We reached this conclusion after several speculative steps. Since the difference in the active ratios between the two graphs at the right is fairly large, a rule directly reflecting this characteristic was expected. However, the

current implementation of DISCAS failed to detect it. A more efficient and effective search of the lattice is desirable in order to lighten the workload of analysts.

4. Concluding remarks

The rules derived by a new version of DISCAS software were interpreted, and we were able to derive valuable hypotheses about the characteristic structures that result in dopamine antagonist activity. The insight into the interpretation process showed that the conditions of a rule alone do not provide enough material to invoke reactions by an active user. The analysts needed a variety of additional information on collateral correlations, relative rules, and ridges, and needed to be able to visualize distributions and browse the supporting instances of a rule. Furthermore, analysts need some knowledge to understand rules in a reasonable way. The analyst with hypotheses related to the problem before inspecting the rules will be rewarded much more.

There are still points to improve. The selection of attributes should include common functional groups. The previous section also did not discuss the difficulties in the interpretation of preconditions. If a precondition excludes a well-defined subgroup of compounds, its meaning will be clear. However, there are often 3 or 4 precondition clauses, and the overlap of excluded instances makes interpretation difficult. One possible reason for this difficulty is the lack of collateral correlation information for the preconditions, which are expected to guide the survey as those for the main condition did. We are going to implement these functions in the forthcoming version of DISCAS.

Acknowledgements

The author thanks Dr. Masumi Yamakawa for his work on dopamine antagonist analysis, and for his discussion of the inspection process.

References

[1] T. Okada, “Discovery of Structure Activity Relationships using the Cascade Model: The Mutagenicity of Aromatic Nitro Compounds”, *Journal of Computer Aided Chemistry*, Vol. 2, pp.79-86 (2001).

[2] T. Okada, “Characteristic Substructures and Properties in the Chemical Carcinogenicity Studied by the Cascade Model”, *Proc. International Workshop on Predictive Toxicology Challenge 2001*, PKDD-2001, Freiburg (2001).

[3] T. Okada, “Efficient Detection of Local Interactions in the Cascade Model”, In T. Terano et al. (Eds.) *Knowledge Discovery and Data Mining PAKDD-2000*, Springer-Verlag, LNAI 1805, pp.193-203 (2000).

[4] T. Okada, “Datascape Survey using the Cascade Model”, to appear in K. Satoh et al. (eds.) *Discovery Science 2002*, Springer-Verlag (2002).

[5] T. Okada, “Topographical Expression of a Rule for Active Mining”, In H. Motoda (ed.) *Active Mining*, pp.247-257, IOS Press (2002).

[6] T. Okada, and M. Yamakawa, “Mining Characteristics of Lead Compounds using the Cascade Model” (*in Japanese*), to appear in *Proc. 30th Symposium on Structure-Activity Relationships*, Toyohashi (2002).

Extracting Geographical Knowledge from the Internet

Ourioupina Olga
Saarland University
ourioupi@coli.uni-sb.de

Abstract

This paper describes an algorithm for knowledge acquisition in the Geography domain. We apply Text Mining procedures to the Internet in order to classify places into different location types (e.g., Maebashi is a CITY, Honshu is an ISLAND) and to determine for a given place name, where the place is (e.g. Maebashi is in Japan, Honshu is in the Pacific ocean). At the moment we distinguish 6 location types.

We conducted three series of experiments: with a manually tuned system, using the TiMBL Memory Based Learner, and using the C4.5 Decision Tree Induction Algorithm. The results obtained so far are quite promising: all three algorithms scored in the high eighties for all but one (CITY) class. All systems were significantly better than the baseline. That leads us to the conclusion that the approach may successfully be used to automatically create gazetteers for Named Entities Recognition tools.

1. Introduction

Many linguistic tasks, such as, for example, Coreference Resolution or Information Extraction/ Retrieval require some knowledge about proper names. This knowledge comes normally from Named Entities (NE) Recognition systems.

Most NE Recognition algorithms make extensive use of gazetteers — huge lists of names of locations, people, etc. Creating such lists by hand is a very time consuming task. Therefore, some researchers simply downloaded available lists for each category from several web sites. This was done for the MENE system [4], for example. However, when not manually checked, these lists are far from being exhaustive and do not provide full classification. And for most languages those lists do not exist at all. As an alternative, many researchers have tried to make NE systems robust without relying on the availability of gazetteers at all. An example is the system developed by A. Mikheev et al. and described in [11], which can identify names of organisations and people relatively well (85,5% and 92,4% F-measure respectively) even without gazetteers. Locations, however, were much worse (51,7% F-measure without gazetteers compared to 94,5% with gazetteers).

Gazetteers seem to become much more usefull for the tasks, where fine-grained classification of NEs is needed. In fact, the commonly used NE classification (PERSON, ORGANIZATION, LOCATION, TIME, MONEY, and PERCENT), developed for the Message Understanding Conferences (MUC), is often said to be too rough. For example, in the Textract Information Extraction system [16] (the best system in one of the two TREC-8 Question Answering tasks), the MUC scheme was significantly changed: more classes (e.g., PRODUCT) and subclasses (e.g., SCHOOL as a subtype of ORGANISATION) were added. The authors claim that "an extended NE tagset is helpful for sophisticated Information Extraction and Question Answering".

We believe that Data Mining techniques can be used successfully to acquire exhaustive fine-grained gazetteers automaticaly and thus avoid hand-coding. This paper describes an algorithm for subclassifying names of locations. The algorithm can be used offline (to produce a gazetteer) or online (as an alternative to gazetteers). The algorithm can be easily extended to other NE classes (such as organisations and persons), providing sufficient information for NE Recognition tools.

Additionally, the same approach can be used for determining the location of a given place name, as described in Section 7. This information can be used for modelling the Geography domain, for example, for creating domain specific dictionaries and taxonomies.

2. Related work

Extensive research on pattern extraction from raw text has already been done. For example, M. Hearst in [8] developed an approach for discovering lexico-syntactic patterns for hypernyms. Several papers, such as [12] and [13], described ways of knowledge acquisition for various domains.

However, all these systems use limited amounts of texts as input data. For our task we need a much bigger corpus, because most names of locations are infrequent and, therefore, unlikely to be found in a conventional corpus. This problem can be solved, if one uses the largest data set possible, which is, obviously, the World Wide Web.

Most papers on information extraction from the Internet, such as [3] and [15] focus on algorithms, requiring extensive processing of many web pages. These

approaches are not acceptable in our case for two reasons: first, due to the lack of resources, and second, because of the time requirements — our main goal is to use the algorithm for the online classification.

This leads us to a conclusion that we have to restrict ourselves to very simple data types that can be obtained from the Internet quickly, such as the number of pages returned by a search engine for a given query. F. Keller et al. have shown recently in [9] that web frequencies for bigrams correlate with frequencies obtained from corpora and with human plausibility judgements. That makes us believe that web frequencies are good predictors for the behaviour of words and word combinations.

3. Data

Our system subclassifies names of locations. At the moment, the following classes are distinguished: CITY, REGION, COUNTRY, ISLAND, RIVER, MOUNTAIN. However, incorporating additional classes is not problematic. As the classes may overlap (for example, "Washington" belongs to the classes CITY, REGION, ISLAND and MOUNTAIN), the problem was reformulated as six binary classification tasks.

Our main dataset consists of 1260 names of locations. Most of them were sampled randomly from the indexes of the World Atlases [1], [5], and [10]. However, this random sample contained mostly names of very small and unknown places. In order to balance it, we added a list of several countries and well-known locations, such as, for example, Tokyo or Hokkaido. Finally, our dataset contains about 10% low-frequency names (<20 Web pages pro name), 10% high-frequency names (>1000000 pages pro name, the most frequent one (California) was found by Altavista in about 25000000 pages), and 80% medium-frequency ones.

These names were classified manually using the above mentioned atlases and the Statoids webpage [2]. An example of the classification is shown in table 1.

The dataset was used in two different ways. First a quarter of the data was used to create the manually tuned system described in Section 4. Then the same subset was used to train a Memory Based Learner and a Decision Tree induction algorithm. In another experiment, the whole original dataset was provided to the same Machine Learning systems.

Table 1. Gazetteer example

Maebashi	CITY
Magdalena	CITY, RIVER, ISLAND, REGION
Magwe	CITY, REGION

4. Algorithm

For each class we constructed a set of patterns. All the patterns have the form "KEYWORD+of+X" and "X+KEYWORD". Each class has from 3 (ISLAND) up to 10 (MOUNTAIN) different keywords. For example, for the class CITY we use 4 keywords ("city", "town", "mayor", "streets") and 7 corresponding patterns ("city+of+X", "X+city", "town+of+X", "mayor+of+X", "X+mayor", "streets+of+X", and "X+streets"; note that "X+town" is not included, because it does not provide reliable counts). Keywords and patterns were selected manually: we tested many different candidates for keywords, collected counts (cf. bellow) for the patterns associated with a given candidate, then filtered most of them out using the t-test. The remaining patterns were checked by hand. In future we plan to apply bootstrapping techniques in order to select keywords and patterns automatically.

For each name of location to be classified, we construct queries, substituting this name for the X in our patterns. We do not use morphological variants here, because morphology of proper names is quite irregular (compare, for example, the noun phrases *Fijian* government and *Mali* government). Then the queries are sent to the AltaVista search engine. The number of pages found by AltaVista for each query is then normalised by the number of pages for the item to be classified alone (the pattern "X", without keywords). If several of these counts exceed certain predefined thresholds, we classify the item as [+CLASS], otherwise as [-CLASS]. Thresholds vary across classes and patterns. Consider an example for the item "Maebashi" and the class CITY:

Table 2. Queries and counts for "Maebashi"

Queries	Alta Vista counts	Normalised counts	Threshold for the pattern
"Maebashi"	4887		
"Maebashi+city"	925	0.19	0.001
"city+of+Maebashi"	13	0.003	0.00083
"town+of+Maebashi"	—		
…	—		

For the class CITY we need only two counts above the thresholds. So, when two counts exceeding the thresholds (for "Maebashi+city" and "city+of+Maebashi") are found, the system stops, classifies "Maebashi" as [+CITY] and does not send more queries. All the parameters (thresholds for counts for each pattern and number of counts exceeding the thresholds needed for each class) are tuned manually.

Although the algorithm works relatively well, it suffers from the data sparseness problem, when classifying less

frequent names. Therefore additional procedures were added to deal with rare words. They are applied only if an item was not classified by the main algorithm as belonging to at least one class.

At the first step, the thresholds for the parameter "number of counts required" are relaxed for each class: an (infrequent) item is classified as [+CITY], for example, if at least one pattern for the class CITY provides a good count. If that does not help and the item remains unclassified, we use additional patterns, containing much more frequent keywords than "city", "mayor", etc. We tried various locative and some other prepositions as possible candidates for keywords, resulting in such patterns as, for example, "along+X", "along+the+X", "against+X". The results, however, were discouraging: all the items that could possibly be classified by this subprocedures, had already been classified at one of the two previous steps. The only additional pattern that turned out to be useful was "the+X": an item is classified as [+CITY, –ISLAND, –RIVER, –REGION, –STATE, –MOUNTAIN], if it is used only rarely with the definite article, as [–CITY, –ISLAND, +RIVER, –REGION, –STATE, –MOUN-TAIN], if it is used very often with the definite article, and as [–CITY, –ISLAND, –RIVER, –REGION, –STATE, –MOUNTAIN] in the intermediate case. The final algorithm is summarised below:

```
For an item X to be classified:

N=get_count_from_Altavista("X");

foreach C { //class
  found(C)=0;
  foreach p {  //precompiled pattern
    q=construct_query(p,X);
    n=get_count_from_Altavista(q);
//T(p,C),T(C) - predefined thresholds
    if (n/N>T(p,C)) found(C)++;
    if (found(C)>T(C)) {
      classify(X,C);
      last;
    }
  }
}

if (unclassified(X)&rare(X)) {
  foreach C {
    if (found(C)>T(C)-1) classify(X,C);
  }
}

if (unclassified(X)&very_rare(X)) {
  n=get_count_from_Altavista("the+X");
//T1, T2 - predefined thresholds
  if (n/N>T1) classify(X,RIVER);
  if (n/N>T2) classify(X,CITY);
}
```

We used about a quarter of our data to develop the system and tune all the parameters. The remaining data were reserved for testing. Table 3 shows test results in two modes: only main algorithm vs. all the procedures described above. Accuracy is counted as the number of correctly assigned [+CLASS] and [–CLASS] labels divided by the size of the test set. As a baseline we used the [+CITY, –ISLAND, –RIVER, –REGION, –STATE, –MOUNTAIN] assignment for all the items.

Table 3. System's accuracy for different classes

Class	Baseline	Main algorithm	All the algorithms
ISLAND	84.6%	93.6%	94.2%
CITY	55.8%	66.1%	75.3%
REGION	79.6%	87.9%	87.6%
COUNTRY	86.3%	99.4%	99.4%
RIVER	82.0%	89.4%	89.2%
MOUNT	85.6%	87.9%	87.5%
Average	77.0%	87.4%	88.9%

5. Using Machine Learning

For our second and third experiments we used the TiMBL Memory Based Learner and the C4.5 Decision Tree Induction algorithm, described in [7] and [14] respectively. Memory Based Learning (MBL) is a lazy learning method. It keeps all the training data in memory and does not make any abstractions or restructuring. As pointed in [6], MBL is very helpful for linguistic tasks, because exceptions and sub-regularities are treated more accurately than by standard Machine Learning techniques. The C4.5 algorithm, on the contrary, prunes trees extensively. However, it has a very important advantage: Decision Trees approach allows us to use less patterns for each item, therefore to send less queries to Altavista and, as a result, increase processing speed.

To start with, we trained TiMBL and C4.5 with our development set (exactly the same items that were used to create the system described in Section 4). Then we used the remaining data for the testing, thus, replicating our first experiment (with the manually tuned system).

The following features were included: count for the item (pattern "X"), counts for queries (as described above), and the same counts normalized by the count for the item (for comparison, in our manually tuned system only the normalised counts were used). We performed two different training/testing runs: first, we used all the features for all the classes. In the second run, we included for each class only those patterns that were used for this class in our manually tuned system. Possible way to reduce the number of features is discussed in section 6. For example, for the class CITY we used the following features (#Y stands for the number of web pages returned by the Altavista search engine for the query Y):

Table 4. Features for TiMBL and C4.5, class CITY

Used for both runs (8 queries, resulting in 15 features)	#X #("city+of+X") #("X+city") … #("streets+of+X") #("X+streets")	#("city+of+X")/#X #("X+city") /#X … #("streets+of+X") /#X #("X+streets") /#X
Used for the first run only (37 queries, 74 features)	#("island+of+X") #("X+islands") …	#("island+of+X") /#X #("X+islands") /#X …

Table 5 summarizes the results of these two runs. For the C4.5 algorithm we show only the accuracy after the pruning. Before the pruning, system performed on average 0.5-1% worse.

Table 5. Accuracy of TiMBL and C4.5 on the test data (same as in table 3)

Class	Base-line	MBL, all the features	MBL, prese-lected feat.	C4.5, all the feat.	C4.5, prese-lected feat.
ISLAND	84.6%	92%	94.5%	92.8%	92.4%
CITY	55.8%	64.4%	63%	66.3%	62%
REGION	79.6%	83.5%	83.7%	88.1%	88.2%
COUNTRY	86.3%	99.4%	97.1%	98.1%	97.5%
RIVER	82.0%	87.2%	86.4%	86.5%	89.4%
MOUNT	85.6%	58.8%	76.8%	68.7%	86.6%
Average	77.0%	80.9%	83.6%	83.4%	86.0%

For our last experiment we used the whole dataset and performed 10-fold cross validation and leave-one-out testing (system is trained on all the items but one, then tested on the remaining item; the same is done for all the items in turns and the results are summed up). The accuracy is shown in tables 6 and 7.

Table 6. Accuracy of MBL and C4.5 in the 10-fold cross-validation test

Class	MBL, all the features	MBL, prese-lected features only	C4.5, all the feat.	C4.5, prese-lected feat. only
ISLAND	91-97%	89-97%	91-99%	92-98%
CITY	67-77%	63-74%	75-83%	70-78%
REGION	83-89%	81-90%	83-89%	83-94%
COUNTRY	97-100%	96-98%	96-99%	96-99%
RIVER	79-90%	84-91%	81-94%	87-94%
MOUNT	81-91%	83-91%	83-92%	83-89%
average	87.3%	86.4%	88.7%	88.5%

Table 7. Accuracy of MBL and C4.5 in the leave-one-out test

Class	MBL, all the features	MBL, prese-lected features only	C4.5, all the feat.	C4.5, prese-lected feat. Only
ISLAND	93.0%	93.1%	93.1%	94.0%
CITY	73.7%	68.4%	78.4%	73.8%
REGION	87.4%	86.4%	87.9%	89.8%
COUNTRY	97.9%	97.5%	97.9%	98.5%
RIVER	86.2%	86.5%	89.3%	91.0%
MOUNT	87%	86.8%	87.8%	86.2%
average	87.5%	86.5%	89.1%	88.9%

6. Discussion

First, all the classes, except CITY, were resolved by our manually created system with an accuracy of about 90%. And most mistakes for the class CITY were due to infrequent words (that are unlikely to be found in any real application). That leads us to the conclusion that the system can successfully be used to create gazetteers: we can replace uncontrollably large amount of hand-coding for gazetteers by relatively little work on tuning the system (based on a very small gazetteer).

Memory Based Learning and C4.5 did not achieve the same accuracy when trained on a very small gazetteer. MOUNTAINs came out even worse than the baseline algorithm. However, when a bit more training data were presented, MBL performed only slightly worse than the manually tuned system: all the differences between the results of 10-folds cross-validation and the performance of hand-tuned system were nonsignificant. C4.5 performed nearly the same as the manually tuned system and was almost always better than MBL, but the difference was not significant either. The good Machine Learning results show that we can have a fully automatic system that presupposes no manual intervention except for the initial classification of the tuning data.

When one wants to use the system online for classifying items in real time a second issue becomes important. In that case the number of queries sent to AltaVista plays a very important role: each query slows the system down dramatically. Our manually tuned system sends on average less queries than both (all vs. preselected features) versions of the MBL-based algorithm, because it stops when enough "good counts" are collected. If one needs the full classification (all the 6 classes), then the "all features" and "preselected features" systems send exactly the same number (38) of queries (every query from the "all features" variant is used for at least one of the six classification tasks in the "preselected features" variant). However, if we are interested only in

some classes (e.g., whether an item is REGION, COUNTRY or none of these two), the "preselected features" version becomes more practical. Also, the MBL and the C4.5-based systems both perform better with preselected features, when only very few train data are available and these data do not represent some classes well enough (cf., for example, accuracies for MOUNTAIN in table 5).

Compared to the manually created system, the C4.5-based algorithm is quicker only in the case of feature preselection: both the manually created system and C4.5 with all the features send about 7 queries per class in the worst case. However, when the preselected features are used, the C4.5-based algorithm sends only 5 queries pro class, although the accuracy is almost the same.

7. Determining locations

We applied the methodology, described above, to another task: given a place name, determine where this place is located. This information can be used by inference tools in the Geography domain.

To start with, we conducted a very simple preliminary experiment: given an island(s) name, determine the ocean where this island is.

We used all the island names from our initial 1260-words dataset and annotated them with the corresponding ocean names. Finally our data consisted of about 200 islands. Although several names in the dataset corresponded to different islands in different oceans, we formulated our problem as a single classification task with 4 classes: ARCTIC, ATLANTIC, INDIAN, and PACIFIC. Due to this task formulation, the maximal theoretically possible accuracy on this data set was 98%.

The following patterns were used for this task: X+ AND+OCEAN, "X+island"+AND+OCEAN, "X+ islands"+AND+OCEAN, where OCEAN stands for "Arctic", "Atlantic", "Indian", and "Pacific". As the features we used the counts for the corresponding queries and the same counts, normalised by OCEAN's frequencies (i.e., counts for "Arctic",…,"Pacific" respectively). First we tried all the features one by one. Then we combined them in the following manner: if #"X island" was bigger than #"X islands", then we choose the OCEAN that has the highest "X+island"+AND+OCEAN count. Otherwise we choose the OCEAN that has the highest "X+islands"+AND+OCEAN count. If both counts are small, we take #(X+AND+OCEAN). If #X is too big, we take #(X+AND+OCEAN)/#OCEAN instead. Table 8 shows example counts for the items "Tokelau" and "Camaguey".

For "Tokelau" our system finds out first that this item is more probably a group of islands, than a single island. Therefore, it compares counts for "X+islands"+AND +OCEAN and picks up the ocean with the highest count.

Table 8. Sample runs for "Tokelau" and "Camaguey"

Queries	Simple counts	Normalised counts
"Tokelau+island"	563	
"Tokelau+islands"	1456	
"Tokelau+islands"+AND+Arctic	38	—
"Tokelau+islands" +AND+Atlantic	836	—
"Tokelau+islands" +AND+Indian	928	—
"Tokelau+islands" +AND+Pacific	1121	—
"Camaguey+island"	0	
"Camaguey+islands"	1	
Camaguey	4003	
Camaguey+AND+Arctic	40	6.79e-5
Camaguey+AND+Atlantic	340	1.16e-4
Camaguey+AND+Indian	309	4.34e-5
Camaguey+AND+Pacific	353	4.23e-5

As a result, "Tokelau" is classified as located in the Pacific ocean. For "Camaguey" the system cannot use the same strategy due to lack of data. Therefore, counts for X+AND+OCEAN are compared. However, the word "Camaguey" is frequent enough, so that the system should normalize the counts. Finally, "Camaguey" is correctly classified as located in the Atlantic ocean.

Testing results are summarised in table 9.

Table 9. Accuracy for the Islands/Oceans task

Maximum possible	98%
Worst feature	55%
Best feature	71%
Combined features	77%

We also tried to apply Memory Based Learning algorithms to the same data set, using the same features, as for the manually tuned system. However, the results were not promising, due to lack of data. Table 10 shows the accuracy on the 4-fold evaluation and leave-one-out test. In the future, we want to collect more data and come back to this problem: although the results are not good enough, MBL does better than the worst feature.

Table 10. MBL Accuracy for the Islands/Oceans task

4-fold evaluation	48-80%
leave-one-out	60%

8. Conclusion and future work

We described several experiments on acquiring geographical knowledge from the Internet. The results can be used, on the one side, by Named Entity Recognition systems, and, on the other side, to provide

material for inference tools, aimed at reasoning in the Geography domain.

Our experiments show that simple Data Mining techniques may help to create gazetteers automatically and thus avoid hand-coding. With a small amount of data (about 1000 items) we were able to train the TiMBL Memory Based Learner and the C4.5 Decision Trees Induction algorithm, achieving performance compatible with the manually tuned system and significantly better than the baseline.

In the future we plan, on the one hand, to incorporate more classes (for example, SEA). On the other hand, we want to extend the system to cover all the major types of proper names and not only locations (for example, names of organisations, people, products, etc will be included). When this is done, we can start experimenting with NE Recognition algorithms in order to see, how good or bad NE Recognition works with automatically created gazetteers, compared to hand-coded and no gazetteers cases.

We also plan to use other machine learning techniques and try to improve the performance using co-training.

Another important plan concerns bootstrapping: we want to use mutual bootstrapping in order to, on the one hand, extract lists of items to be classified, and, on the other hand, improve our keyword sets. That would allow us to create domain-specific lexicons even more automatically, starting with only a few seed patterns. Additionally, improvements in the keyword selection may help to decrease the number of queries to be sent.

As far as our second task is concerned, we are currently performing more complicated experiments, trying to link classified items to each other. For example, once we classified some items as cities, some as regions and some as countries, we may want to find out automatically, whether some of our cities and regions are located in our countries, and, if yes, in which country exactly.

9. References

[1] *Atlas of the World,* George Philip Limited, London, 2000.

[2] *Administrative divisions of countries ("statoids")*, http://www.mindspring.com/~gwil/statoids.html.

[3] S. Brin, Extracting Patterns and Relations from the World Wide web, *WebDB Workshop at EDBT '98*, Valencia, 1998, pp. 172-183.

[4] A. Borthwick, J. Sterling, E.Agichtein, and R. Grishman, Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition, *Proceedings of the Sixth Workshop on Very Large Corpora*, Montreal, 1998.

[5] *Collins New World Atlas*, HarpersCollinsPublishers, London, 2001.

[6] W. Daelemans, A. van den Bosch, and J. Zavrel, Forgetting Exceptions is Harmful in Language Learning, *Machine Learning, Special Issue on Natural Language learning* 34, Kluwer Academic Publishers, Dordrect/Boston/London, 1999, pp. 11-41.

[7] W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch, *TiMBL: Tilburg Memory-Based Learner*, ILK Technical Report – ILK 02-01, Tilburg, 2002.

[8] M. Hearst, Automatic acquisition of hyponyms from large text corpora, *Proceedings of the Fourteenth International Conference on Computational Linguistics*, GETA(IMAG) & Association Champollion, Nantes, 1992, pp. 539-545.

[9] F. Keller, M. Lapata, and O. Ourioupina, Using the Web to Overcome Data Sparseness, Proceedings of the Conference on Empirical Methods in Natural Language Processing, Philadelphia, 2002, pp. 230-237.

[10] *Knaurs Atlas der Welt*, Droemer Knaur, München, 1994.

[11] A. Mikheev, M. Moens, and C. Grover, "Named Entity Recognition without Gazetteers", *Proceedings of the Ninth International Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, Bergen, 1999.

[12] D. Moldovan, R. Girju, and V. Rus, Domain-Specific Knowledge Acquisition from Text, *Proceedings of the Applied Natural Language Processing (ANLP-2000) conference*, Morgan Kaufmann Publishers Inc, San Francisco, 2000, pp. 268-275.

[13] W. Phillips and E, Riloff, Exploiting Strong Syntactic Heuristics and Co-Training to Learn Semantic Lexicons, Proceedings of the Conference on Empirical Methods in Natural Language Processing, Philadelphia, 2002.

[14] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc, San Francisco, 1993.

[15] S. Soderland, Learning to Extract Text-based Information from the World Wide Web, *Proceedings of Third International Conference on Knowledge Discovery and Data Mining*, 1997, pp. 251-254.

[16] R. Srihari and W. Li, Information Extraction Supported Question Answering, *The Eighth Text Retrieval Conference (TREC-8)*, Department of Commerce, National Institute of Standards and Technology, 1999, pp. 185-196.

VDL: A Language for Active Mining Variants of Association Rules

Kok-Leong Ong Wee-Keong Ng Ee-Peng Lim
Nanyang Technological University, Nanyang Ave, N4-B3C-14
Singapore 639798, SINGAPORE
ongkl@pmail.ntu.edu.sg

Abstract

The popularity of association rules has resulted in several variations being proposed. In each case, additional attributes in the data are considered so as to produce more informative rules. In the context of active mining, different types of rules may be required over a period of time due to knowledge needs or the availability of new attributes. The present approach is the ad-hoc development of algorithms for each variant of rules. This is time consuming and costly, and is a stumping block to the vision of active mining. We argue that knowledge needs and the changing characteristics of the data requires the ability to re-specify the type of rules to rediscover over time. This paper proposes a novel approach to specify the "how-to" of mining different rule variants without the cost of developing new algorithms. Called the VDL, it is SQL-like and has the expressive power demonstrated by our examples, some of which are classical and others novel. We also give a discussion on the theoretical model underpinning our proposal.

1. Introduction

The discovery of association rules has been a popular domain of study in data mining. We call a rule $X \rightarrow Y$, where X and Y are sets of items, a *base* rule [1]. Over the years, different attributes [2, 4, 7, 8, 10, 15] were considered to create more informative rules which we refer to as *variants*. Usually, a variant of the base rule considers additional attributes to produce more informative rules as illustrated in our examples below.

Example 1 Let $X = \{a, d\}$ and $Y = \{e, y\}$ such that X and Y satisfies the support and confidence requirements, then the base rule would be $X \rightarrow Y$ or $\{a, e\} \rightarrow \{e, y\}$. If we consider the number of times an item occurs in X or Y (i.e., the recurrence), we have a more informative rule that may be represented as $\{2\,a, 1\,d\} \rightarrow \{3\,e, 5\,y\}$. We call this rule [15] a variant of the base rule, and it means that two occurrences of 'a' and an occurrence of 'd' leads to three occurrences of 'e' and five occurrences of 'y'.

Example 2 Assuming the same X and Y in Example 1, another variant of $X \rightarrow Y$ may be obtained if we consider their spatial relationship. One such rule would be $\{\texttt{left}(a, d)\} \rightarrow \{\texttt{top}(e, y)\}$ which means that whenever 'a' appears on the left of 'd', 'e' will appear on top of 'y'. This rule is another variant [7] from the base rule, where each transaction contains the spatial relationship of each item with respect to the others.

In the context of active rule mining, the above examples illustrate an important motivation. Data attributes (e.g., the recurrence value in Example 1 and the spatial relationships in Example 2) are continuously added or remove, and knowledge needs changes over time. While the analyst may find the base rules useful today, the inclusion of a new attribute may motivate the need to rediscover rules with the new attribute in mind. A survey of existing literature shows various attributes been considered, and each presents their ad-hoc solution of an algorithm to address the new attribute's contribution to mining. Clearly, the cost of developing algorithms is both costly and time consuming. This is evident with the availability of *only* base rules in most commercial data mining tools. And this is a stumping block towards the vision of active mining where new variants may be needed over time. To address this problem, the ad-hoc approach taken by the current community must be re-evaluated.

Looking at the database industry, we have taken for granted the availability of SQL to retrieve relevant tuples from the database. Before SQL and the relational model, this has been done in an ad-hoc fashion, where individuals write code to retrieve relevant tuples from proprietary data models. This chaotic situation is similar to the current state of data mining for rules. While data mining languages has been proposed, non to our knowledge propose a declarative approach on the "how-to" of finding different rule variants.

Taking the cue from the database industry, our contribution in this paper is to address this issue. The Variant Description Language (VDL) is our effort to declaratively describe different approaches to finding variants of rules using the Apriori [1] as the model to eliminate the need for developing algorithms. This in turn will enable speedy re-specification of rules to mine and hence, achieve active mining of new rules from ever changing data sources.

The remaining sections of the paper is organized as follows. In the next section, we discuss some related work on declarative languages. We then illustrate the expressiveness of our language in Section 3 using examples of which some are classical and others novel. Section 4 then gives a brief description of the theoretical models that the language is built upon. Finally, we summarize our work in Section 5.

2. Related Work

The work in this paper was motivated from Mannila's [9] discussion on a theoretical framework for data mining. He commented on the ad-hoc situation of data mining research and called for a systematic framework to develop KDD applications. A framework for mining rules was discussed but lacked concrete details. Our work continues from where Mannila stopped. Although far from a theoretical foundation, the proposal is a step closer towards a systematic approach of knowledge discovery.

Closest to our proposal is the `MINE RULE` operator proposed by Meo et al [11]. Also SQL-like, the operator enables a uniform and consistent description of the problem of discovering association rules. Using the `MINE RULE` operator, the authors demonstrate how it can be used to describe the different rule mining tasks such as mining base rules with data constraints or rules at multiple concept levels (which is one of the variants known [4]). Of course, the difference lie in the objective of the language. Meo's `MINE RULE` operator is concerned with the uniform description of different but similar tasks of rule mining where each implementation of the algorithm is in place. Although similar in motivation, our proposal describes how different parameters are considered to find different variants of rules. In our case, no algorithms exist, and in place is an algorithmic engine that can, with the VDL, mine future variants.

Around the same time, a more generalized declarative language to describe different data mining tasks was proposed by Han et al [5]. The language is also SQL-like and arises from the **DBMiner** [5] project where different data mining algorithms are implemented. Hence, the Data Mining Query Language or DMQL in short, was created to specify the algorithm (e.g., classification, clustering or association rules), its parameter values (e.g., category labels, number of clusters or support) and the data set to use for a particular data mining task. Thus, DMQL has a similar motivation as Meo's `MINE RULE` operator.

3. VDL By Examples

We begin with the conceptual model needed to describe the task of association rule mining. Let $\mathcal{D}$ be a database of transactions. A transaction $T \in \mathcal{D}$ contains items from the universal of all items $\mathcal{I}$ in $\mathcal{D}$ such that $T \subseteq \mathcal{I}$ and $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ where $i_j, 1 \leq j \leq n$ are known as items. Let $P = \{p_1, p_2, \ldots, p_m\}$ be the universe of attributes associated with each item in T in $\mathcal{D}$. Given an item i_x in a transaction T_y, $T_y.i_x.p_z$ returns the attribute value of p_z of i_x in T_y. Using this object oriented notation, the universe of P is never an empty set and contains at least one attribute called `Label` representing the description of an item. Except for the VDL, the label is always implicitly written in this paper (i.e., an item e means that it has the label 'e').

With the above definitions, we can now discuss the VDL using a practical case to illustrate its relevance to active mining of informative rules. The practical case is the transaction database of a typical supermarket. When a customer buys some products, the whole purchase is mapped to the concept of a transaction and each product is an item. In the beginning, what is stored in each purchase are the unique items bought. Conceptually, the database may looked like the one in Table 1.

TID	Items
1	milk bread butter
2	toothbrush milk bread coke beer
3	diapers bread coke

Table 1. A transaction database.

With the available data, the analyst can obtain a set of base rules. This can be done using existing data mining packages. In our proposal, the same set of base rules would be discovered using the VDL and the algorithmic engine. Thus, the VDL would be the SQL of databases, and the algorithmic engine is the implementation of the Apriori model similar to tables as implementation of the relational model. In the Apriori model, the idea of finding an association rule lies in the discovery of frequent itemsets. A frequent itemset can be obtained by (1) generating potential candidate itemsets that are (2) scanned against the database to (3) collect the support count. Since a pass through the database is expensive, itemsets are (4) pruned if we know that it cannot be frequent in the coming pass. Once the support is collected, an itemset is (5) evaluated to determine if it is frequent. If so, it forms the basis of candidate generation in the next round, and is used to obtain rules satisfying the confidence measure.

3.1. Mining Base Rules

Base on the above model, the role of the VDL is to express the five main steps to determine whether a candidate item-

set would be frequent, and hence become the basis of confidence evaluation. To extract the frequent itemsets for the base rules, the formulation of the VDL will be as follows:

```
GENERATE BaseRules USING
CANDIDATES FROM AprioriJoin(f, Lp)
PRUNE IF EXISTS C - C.c NOT IN Lp
VOTE IF Subset(C, T)
INCREMENT C.Count BY 1
SELECT IF C.Count >= MinSupp
```

A run of the above VDL produces the set of itemsets that are frequent. These itemsets form the basis of rule generation in which the confidence measure is evaluated. Observe that we have consciously omitted the description of rule generation as the idea is similar and paper space is an issue. However, the reader should be able to extrapolate the methodology to rule generation. The `GENERATE` clause attaches the description "BaseRules" to the set of evaluation criteria after it. The `CANDIDATES` clause defines how candidates should be generated. From the above, we see that the algorithmic engine exposes a number of internal objects for VDL manipulation. In this case, `Lp` refers to the set of frequent itemsets found in the previous pass and `f` references the candidate 1-itemsets in the bootstrapping phase and subsequently, are the frequent itemsets of size 1.

The bootstrapping is needed as frequent 1-itemsets are not known initially, and hence `Lp` is empty. The engine first identifies the universe of items $\mathcal{I}$ in $\mathcal{D}$. In the case of Table 1, this is determined to be $\mathcal{I} = \{$milk, bread, butter, toothbrush, coke, diapers, beer$\}$. Since we are in the bootstrapping phase, `f` are the candidate 1-itemsets. This means that all items in $\mathcal{I}$ will be evaluated according to the remaining set of criteria specified in the VDL. At the end of the bootstrapping phase, `Lp` will be properly populated with the frequent 1-itemsets and `f` refers to the frequent 1-itemsets from now on. In Apriori, candidates are scan in a level-wise manner. Hence, once all frequent 1-itemsets are found, the next step is to scan for the frequent 2-itemsets. Notice that the `GENERATE` clause will generate candidate itemsets containing two items each based on the definition of `f` and `Lp` (a formal definition of the `AprioriJoin` is given in Section 4).

To obtain optimum performance, pruning is used to determine if a candidate has the possibility of becoming frequent. If we are sure that the candidate cannot be frequent, then there is no need to waste unnecessary CPU time. This pruning criteria is expressed in the `PRUNE` clause. For finding base rules, the expression prunes the candidate itemset `C` (another object exposed by the engine) if its subset, formed by removing one of the item `c` from `C`, fails to be a frequent itemset in the previous pass. In this case, we know that it fails to satisfy the *a-priori* property, and hence cannot be frequent. Thus, the engine proceeds to collect the support count of the current candidate `C` only if the evaluation in the `PRUNE` clause fails.

If the candidate survives the pruning test, it is then checked against each transaction to determine if it supports the itemset. This is represented by the symbol `T` to represent the current transaction under inspection. If the subset test holds, then the `VOTE` clause becomes `true` and the engine determines how it should increment the support count of the current candidate. Notice that the support count of a candidate is also modelled (similar to items) as an attribute named `Count`, and the increment is computed following the expression after the `BY` clause. In this case, the count increments by `1` for each transaction that declares support for the candidate. At the end of the pass through the database, the data mining engine determines if the itemset is frequent by using the test criteria in the `SELECT` clause. In this case, the VDL states that the candidate should be selected if its support count has surpassed the given minimum support.

Clearly, this approach is preferred over writing code to achieve the same task. The time and resources saved can also be better use by the analyst in other aspects of the KDD process. Of course, the reader may argue that the facilities of finding base rules are available in data mining packages. The sub-sections that follows will better illustrate how the VDL addresses the changing needs of knowledge, and why the VDL approach becomes an effective solution to addressing active mining of relevant and informative rules.

3.2. Mining Weighted Rules

As the holiday season approaches, the marketing manager of the store decides to use data mining to prepare a marketing campaign. When the analyst presents the results, the manager was overwhelmed by the number of rules available. In his opinion (which is also ours), it will take too much time and effort to draft a campaign based on these rules. Within limited time, the manager would like to focus on items of interest such as those under promotion, or items that give a higher profit margin. The smart analyst knows that he is able to accomplish this easily with the VDL.

He first requested the marketing manager to identify the products that he would be interested in gaining insights. He then creates a new attribute call "weight" such that each item has a numerical value to indicate its importance with respect to other items in the database. For example, `milk` may be given a weight of `1.2`, while `toothbrush` has a weight of `0.5`. The two numbers model the manager's preference in knowing rules containing `milk` over rules containing `toothbrush`. The analyst then formulates a new way to mine rules that considers the weight of each item. This formulation is given as follows.

```
GENERATE WeightedRules USING
CANDIDATES FROM AprioriJoin(f,
    S(k-1) - {S(k-1).s
    WHERE S(k-1).s.Count < S(k-1).s.MSB}
VOTE IF Subset(C, T)
INCREMENT C.Count BY C.Count * SUM({C.c.Weight})
```

```
SELECT IF C.Count >= MinSupp
```

In the above, we introduce another variant that ranked the importance of each item relative to the others through the notion of "weights". To our knowledge, this was first explored (expressed above) in [2], and a similar idea based on multiple minimum support was later introduced in [8]. By now, it should be clear that the requirements of the manager can be easily met by writing the VDL. This is where data mining packages fail if support for such consideration is unavailable. From the perspective of active mining, it is thus possible for the analyst to produce the "best fitting" set of rules quickly. This is important as a slow turnaround may impede the use of the insights in a timely manner.

In this VDL, we introduce two additional objects exposed by the algorithmic engine: `S` and `k`. As mentioned in Section 3.1, frequent candidates in the Apriori model are discovered in a level-wise manner. As such, the object `k` represents the current size of the candidate itemsets been investigated. As candidates of size `k` are generated, the algorithmic engine places them into the collection `S(k)`. Hence, `S` represents the collection of itemsets of all sizes, and `S(j)`, `j` $\geq$ `1` refers to the collection of candidates of size `j`.

Interestingly, the notion of weights invalidates the *a-priori* property. To maintain the downward closure property, a measure called *minimum support bound* was introduced. This is a scalar value which we represented as an attribute (i.e., `MSB`) of the itemset. From the view point of the algorithmic engine, it is not concerned with how the `MSB` is computed. In the model, it is simply a value obtained by means of some computable function. This value may be a constant, a value populated via SQL, or complex routines that computes the result. In any case, the engine assumes the availability of this value when the process starts. Compared to the generation of candidates in the base rules, all candidates, except those whose support count is less than its own `MSB` value, are used for candidate generation.

We also introduce the notion of *set* operations in this formulation. When used together with the braces (i.e., {}), the scalar operators such as "`+`" and "`-`" become set operators as demonstrated in the `GENERATE` clause. It is also used to determine the range of variables affected by an operation. For example, the `INCREMENT` clause computes the total weight of each item in the candidate `C`. Without the braces, `C.c.Weight` would simply refer to the weight of one of the item in the candidate at each point of evaluation for the `INCREMENT` clause. Using the brace, the semantic of `SUM` is instilled, and all the weights of the items in `C` are evaluated collectively.

3.3. Mining Recurrent Rules

Suppose the same supermarket upgraded their point-of-sales terminals a year later to include the capability of storing the number of items purchased in each transaction. With this information, the analyst realizes that a more informative version of the base rule can be obtained by considering the quantity of each item as illustrated in Example 1. The inclusion of this new attribute, as a result, motivated the need to reflect a new set of rules that may give the organization better competitive advantage. Intuitively, if the rules are used for target marketing, then we see that the quantity given in a rule will help decide how much of each item should go into a bundle. With the base rules initially, the number of items to include in each bundle is at best a guess from the experience of the marketing manager. To extract rules containing recurrent items, we have the following VDL.

```
GENERATE RecurrentRules USING
CANDIDATES FROM RecurrentJoin(f, Lp)
PRUNE IF EXISTS C - C.c NOT IN Lp
VOTE IF Subset(C, T) AND C.c.Qty <= T.c.Qty
    AND C.c.Label = T.c.Label
INCREMENT C.Count BY
    Min({Floor(T.c.Qty / C.c.Qty)
         WHERE T.c.Label = C.c.Label})
SELECT IF C.Count >= MinSupp
```

Since the quantity of an item is now considered, the way candidates are generated is thus different from that of the candidates generated for the base rules. Using the practical example, we must now consider candidates such as the occurrence of 2 loafs of bread and 3 packets of milk (which the `AprioriJoin` will miss). This consideration is taken into account using `RecurrentJoin` instead. While the pruning condition remains unchanged, the test for transaction support has been modified. The test `C.c.Qty <= T.c.Qty` (of the `VOTE` clause) ensures that the transaction declares support if and only if it has that number of items recorded. As an example, suppose we want to test whether a candidate containing 2 loafs of bread and 2 packets of milk is a frequent itemset in the new database depicted in Table 2. Then transaction T_1 can safely declare support since it has 4 loafs of bread and 6 packets of milk. However, T_2 cannot claim support since it has 3 packets of milk, but only a loaf of bread. In other words, we cannot "create" this candidate from T_2 and thus, T_2 cannot claim support.

TID	Items
1	milk(6) bread(4) butter(1)
2	toothbrush(1) milk(3) bread(1) coke(6) beer(1)
3	diapers(1) bread(1) coke(6)

Table 2. The new database where the number in the bracket is the quantity of that item purchased.

To conclude this section, we would like to point the reader to the `BY` clause of the above VDL. Notice that it is not a trivial constant that increments the support count of a

candidate by 1. Instead, a transaction that supports an itemset may have a different support contribution. Continuing from the earlier example, the candidate contains 2 loafs of bread and 2 packets of milk. This means that we can "create" two such candidates from T_1. That is, we can divide the bread into two sets of 2 loafs and for each set, we can assign two packets of milk with two more available. Thus, the support contribution from T_1 would be 2 instead. This is what has been mathematically modelled in the VDL's BY clause.

3.4. Mining Recurrent Weighted Items

It should be clear that as new attributes are considered, the real needs of the analyst will evolve (and vice versa). As such, active mining requires the use of the VDL to quickly achieve the updated set of rules not possible with incremental algorithms proposed to date. More importantly, the power of a declarative language goes beyond that of describing individual attributes. In fact, the discussion up to this point has been based on existing rule variants where algorithms are in placed. The VDL is thus a "summary" of the existing algorithm's behavior.

Here, we show that the VDL's expressiveness goes beyond describing existing variants. In fact, the formulation below is a novel representation of combining different attributes to produce a set of informative rules without coding. This particular variant is the result of combining the discussions in the preceding sections. The resultant VDL considers the interest of the manager (i.e., Weight), and the quantity (i.e., Qty) of each item in the process of discovery. The formulation below is obtained by observing some rules that allows semi-mechanical construction of the VDL involving variants that were individually described. We have elaborated the steps in [14], and we shall leave it as an exercise for the reader to interpret the VDL.

```
GENERATE RecurrentWeightedRules USING
CANDIDATES FROM AprioriJoin(f,
    S(k-1) - {S(k-1).s
    WHERE S(k-1).s.Count < S(k-1).s.MSB}
VOTE IF Subset(C, T) AND C.c.Qty <= T.c.Qty
    AND C.c.Label = T.c.Label
INCREMENT C.Count BY
    Min({Floor(T.c.Qty / C.c.Qty)
         WHERE T.c.Label = C.c.Label}) *
         SUM({C.c.Weight})
SELECT IF C.Count >= MinSupp
```

So far, the individual examples demonstrate how different variants can be described. We then show, with the above formulation, the expressiveness of the language by creating a novel, but informative variant of association rule through a simple composition of their VDL description. And as we unveil each variant's VDL, we also demonstrate how a declarative approach is useful in the context of active mining. Through the discussion of mining weighted rules, we show how we can reflect the knowledge needs of the analyst, and the mining of rules containing recurrent items illustrates how the evolution of the data influence the discovery of new knowledge. Finally, we conclude with a novel example, where attributes are combined to demonstrate how the proposal can scale towards new knowledge needs, beyond what the ad-hoc approach can deliver.

4. Theoretical Model

In this section, we briefly discuss the theoretical models underpinning the design of the VDL. In particular, we explore the data model representing the database, the algorithmic model that describes the mining methodology, and the semantics of the language.

The data model, representing the database, has three tables. In the first, we have a collection of transactions where each has a unique identifier called the TID and a set of item descriptions similar to Table 1. In the second table, the unique key is a combination of the TID and the item description. Each key, i.e., a $\langle$TID, Label$\rangle$ pair, returns a tuple containing a set of attribute values P (e.g., Weight) as defined in Section 3. Conceptually, the first two table are "read-only" during the execution of the VDL and the third is an auxiliary table where candidates are entered as they are generated. Each unique itemset has an entry (in this table) that contains its set of attribute values (e.g., Count, Qty, MSB). Some of these values are updated when the VDL is executed while others steer the behavior of the data mining engine using the conditions defined in the same VDL.

Based on the above, the VDL formulation determines how the algorithmic engine manipulates the contents of the three tables. As mentioned, this was designed using the Apriori as the basis of our work. The motivation of selecting the Apriori approach comes from our observation that most algorithms proposed for various variants were extension of the Apriori. Hence, it is thus natural and logical to design our proposal from this model. More importantly, we observe that the generality of the Apriori provided room for consideration of new attributes that arises in the future. Such consideration is difficult with other methods.

As a final note, the expressions expressed in the VDL has foundations in discrete mathematics, in particular, first order logic and set theory. Even the functions AprioriJoin and RecurrentJoin can be described mathematically. For example, AprioriJoin can be mathematically written as $\mathtt{f} \times_a \mathtt{Lp} = \{\mathtt{f} \cup \mathcal{X} \mid \mathcal{X} \in \mathtt{Lp} \wedge \forall x_i, x_j \in \mathcal{X},\ x_i.\mathtt{Label} \leq x_j.\mathtt{Label} \wedge (i < j)\}$, and a VDL expressed using mathematical notations is thus possible. By having a mathematical foundation to underpin its design, we ensure consistency in our definitions and in turn, makes implementation of the VDL and the algorithmic engine possible. For example, the expression for each clause in the VDL can be written mathematically as shown below.

```
GENERATE RecurrentRules USING
CANDIDATES FROM f ×_a Lp
PRUNE IF ∃c ∈ C, C − {c} ∉ Lp
VOTE IF C ⊆ T ∧ C.c.Qty ≤ T.c.Qty ∧ C.c = T.c
INCREMENT C.Count BY min({⌊T.c.Qty / C.c_i.Qty⌋ | C.c_i = T.c}_{i=1}^{|C|})
SELECT IF C.Count ≥ MinSupp
```

5. Summary

In this paper, we proposed the VDL as the mechanism to specify the "how-to" of finding frequent itemsets in databases. Specifically, we are interested in the discovery of all frequent itemsets where various specific frequent itemsets (e.g., maximal itemsets) can be derived.

With this approach, we eliminate the need to develop algorithms which are often costly and time consuming to implement. By the time an implementation completes, the value of the knowledge obtain may no longer be relevant or useful. In the competitive economy, knowledge must be constantly updated. Incremental updates of similar knowledge has been relatively well addressed with incremental algorithms [3]. However, the changing needs of relevant knowledge from the same data source cannot be ignored. Data changes as attributes are created or removed, and knowledge needs of an organization also changes with time. In the context of rule mining, this new variants of rules may be needed to reflect these changes. Hence, a method to specify the mining of new rules will be required.

The approach proposed in this paper uses the Apriori algorithm as the conceptual model to finding association rules. The Apriori model is simple to understand and is sufficiently expressive for various variants of rules and their compositions. While we have demonstrated a few in this paper, we have actually addressed several others in [14] which has not been reflected in this paper due to the lack of space. The positive aspects of the Apriori model aside, many readers may be concerned with the performance of the data mining engine. Compared to newer methods such as the `FP-Tree` [6], the Apriori solution appears to be unsuitable. Fortunately, this is not the case.

Favoring the Apriori model, it is easier to write the VDL for expressing several known variants which uses the Apriori algorithm as the basis of extension. At the same time, the model supports a more general approach allowing more complex rules to be specified and new attributes, never considered before, to be included. Our experience to use the `FP-Tree` as the basis of such works proved to be unnecessarily complicated and futile [13]. This is due to the high degree of optimizations made in favor of generality. To leverage performance to the level of `FP-Tree` without loosing generality, we have implemented a data structure call `T-Graph` [12] which builds a "transaction graph" representing the database in a compact manner. Using `T-Graph`, the VDL needs to scan only a subset of the database and a part of each transaction. Our experimental benchmarks show that `T-Graph` is as fast as `FP-Tree` without giving up the simplicity of the Apriori model described in this paper.

References

[1] R. Agrawal and R. Srikant. Fast Algorithm for Mining Association Rules. In *Proc. of the 20th Int. Conf. on Very Large Databases*, pages 487–499, Santiago, Chile, Aug. 1994.

[2] C. H. Cai, A. W. C. Fu, C. H. Cheng, and W. W. Kwong. Mining Association Rules with Weighted Items. In *Proc. of Int. Database Engineering and Applications Symp.*, 1998.

[3] D. W. Cheung, J. Han, V. T. Ng, and C. Y. Wong. Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique. In *Proc. of the 12th Int. Conf. on Data Engineering*, pages 106–114, New Orleans, Louisiana, USA, Feb. 1996.

[4] J. Han and Y. Fu. Discovery of Multiple-Level Association Rules from Large Databases. In *Proc. of the 21th Int. Conf. on Very Large Databases*, Zurich, Swizerland, 1995.

[5] J. Han, Y. Fu, W. Wang, K. Koperski, and O. S. Zaine. DMQL: A Data Mining Query Language for Relational Databases. In *SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, Canada, 1996.

[6] J. Han, J. Pei, and Y. Yin. Mining Frequent Pettern Without Candidate Generation. In *Proc. of Int. Conf. on Management of Data*, Dallas, TX, May 2000.

[7] K. Koperski and J. Han. Discovery of Spatial Association Rules in Geographic Information Databases. In *Proc. of the 14th Int. Symp. on Large Spatial Databases*, Maine, 1995.

[8] B. Liu, W. Hsu, and Y. Ma. Mining Association Rules with Multiple Minimum Supports. In *Proc. of the 5th Int. Conf. on Knowledge Discovery and Data Mining*, San Diego, CA, USA, Aug. 1999.

[9] H. Mannila. Methods and Problems in Data Mining. In *Proc. of the Int. Conf. on Database Theory*, Delphi, Greece, Jan. 1997.

[10] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering Frequent Episodes in Sequences. In *Proc. of the 1st Int. Conf. on Knowledge Discovery and Data Mining*, Montreal, Canada, Aug. 1995.

[11] R. Meo, G. Psaila, and S. Ceri. A New SQL-like operator for Mining Association Rules. In *Proc. of the 22th Int. Conf. on Very Large Databases*, Mumbai (Bombay), India, 1996.

[12] K.-L. Ong, W.-K. Ng, and E.-P. Lim. A Framework for Efficient Scalable Mining of Rule Variants. CAIS Technical Report, Nov. 2001.

[13] K.-L. Ong, W.-K. Ng, and E.-P. Lim. Mining Multi-Level Rules with Recurrent Items Using FP'-Tree. In *Proc. of the 3rd Int. Conf. on Information, Communications and Signal Processing*, Singapore, Oct. 2001.

[14] K.-L. Ong, W.-K. Ng, and E.-P. Lim. Mining Variants of Rules Using the CrystalBall Framework. In *Proc. of the 14th Australasian Database Conference*, Adelaide, Australia, Feb. 2003.

[15] O. R. Zaiane, J. Han, and H. Zhu. Mining Recurrent Items in Multimedia with Progressive Resolution Refinement. In *Proc. of Int. Conf. on Data Engineering*, San Diego, 2000.

CrystalClear: Active Visualization of Association Rules

Kian-Huat Ong Kok-Leong Ong Wee-Keong Ng Ee-Peng Lim
Nanyang Technological University, Nanyang Ave, N4-B3C-14
Singapore 639798, SINGAPORE
ongkl@pmail.ntu.edu.sg

Abstract

Effective visualization is an important aspect of active data mining. In the context of association rules, this need has been driven by the large amount of rules produced from a run of the algorithm. To be able to address real user needs, the rules need to be summarized and organized so that it can be interpreted and applied in a timely manner. In this paper, we propose two visualization techniques that is an improvement over those used by existing data mining packages. In particular, we address the visualization of "differences" in the set of rules due to incremental changes in the data source. We show that visualization in this aspect is important to active data mining as it uncovers new insights not possible from inspecting individual data mining results.

1. Introduction

The discovery of association rules has been a popular domain of study. An association rule [8] is a rule of the form $X \rightarrow Y$, where $X, Y \subset \{i_1, i_2, \ldots, i_j\}$ such that $X \cap Y = \varnothing$ and $i_k, 1 \leq k \leq j$ is an item in the transaction database. A rule is deemed to be of interest to a domain analyst if it satisfies two basic measures of interestingness — support and confidence. However, the simplicity of such measures often resulted in too many rules been produced from a run of the data mining algorithm. While increasing the support and confidence level reduces the rule count, the consequence is the loss of important rules that have low support in the database [16]. To overcome this, additional measures has been proposed [9, 15] to prune away rules that has been objectively identified as having no contributions to insights. Effectively, this helps the analyst focus on rules that might be useful. However, this approach has its own pitfalls.

First, the use of additional measures at best achieves a reduction on the number of rules. This, in essence, does not help the analyst identify the important insights quickly due to its presentation of the results – often in unorganized raw textual form. Second, the pruning of rules may be minimal and huge number of rules (in magnitude of hundreds) may remain in the result. And going through these rules remain a daunting and time consuming task that suggest the need for further summarization and organization. Third, the use of interestingness measures is a double edge sword. If used incorrectly, the measures may produce no results or worse, point to a wrong set of insights [18]. This is indeed unavoidable if the knowledge worker lacks sufficient training or technical know-how.

Hence, visualization forms the posterior step after the results of data mining. It's objective, similar to interestingness, aims to present the results effectively so that an analyst can take advantage of visual cues to help sieve through insights that are otherwise difficult in textual form. Unfortunately, our survey of the existing visualization techniques reveals that this aspect of research has been weakly addressed. In the context of active mining, this is an important issue. In the real world where data condition changes and user needs evolve over time, each step of the KDD process must be leveraged to cooperatively achieve such a goal. In the narrower context of active rule mining, we see incremental algorithms been proposed [13, 14] to address the iterative needs of updated rules. In the same way, the interestingness measures have served well in helping analyst focus on important rules. This brings forth the question: Is the current state-of-the-art on visualization on par with the needs of active mining?

The answer to the above question is what motivated this paper. Having observed the lack of such capabilities, our contributions are as follows.

- We first survey the current state-of-the-art in visualization of association rules in Section 2 to gain an appreciation on the current limitations.
- We then propose an improved visualization technique that uses color cues and spatial organizations to enable a controlled and selective view of the rules obtained

from data mining. Two techniques are proposed and discussed in Section 3.

- From this improved visualization, we consider the evolution of data which in turn, induces an evolving sets of rules where new ones are created, old ones removed, and others changed. This change itself carries knowledge that are important, and visualization is the best candidate for highlighting such insights. We discuss this in Section 4.

2. Related Work

Visualizing rules graphically is a depiction of one-to-one, many-to-one or many-to-many mapping of information items. Prior works on presenting the results of association rule mining can be generally summarized into four common techniques: 2-dimension matrices, directed graphs, tables and grids. Among them, the objective is to represent, graphically, the parameters in association rules namely, the set of antecedent and consequent items, their associations, the support and confidence. In this paper, we briefly discuss these techniques in terms of their strength, weaknesses and tools using such methods.

Two Dimension Matrices The basic design of a two dimension matrix [1, 7, 5] positions the antecedent and consequent items on the X and Y axis respectively. Using customized icons drawn on the matrix tiles, the association between the antecedent and consequent items are identified. Here, different icons can be used to represent support and confidence values. The strength of such visualization is the display of one-to-one binary relationships, where the rule is simple. However, the matrix approach breaks down when there is a need to investigate many-to-one (i.e., rules with multiple antecedent items) or many-to-many (i.e., multiple items in both the antecedence and consequence) relationships. As shown in Figure 1, representing beyond one-to-one relationships using a matrix is actually confusing to the analyst. As a result, it becomes a weak candidate when rules are never always one-to-one.

Directed Graphs A directed graph [2, 17, 7, 6, 10, 11] overcomes the problem in a two dimension matrix. Each node in the graph represents an unique item. An edge connecting two nodes in the graph represents an association. While it has the merits of displaying different relationship types, it is good for cases where only a few items and edges are involved. With many rules, such representation can quickly turned into an entangled display making comprehension difficult. Even if it is possible to layout and display all elements, the limited screen estate makes it next to impossible when following an edge from one node to the other

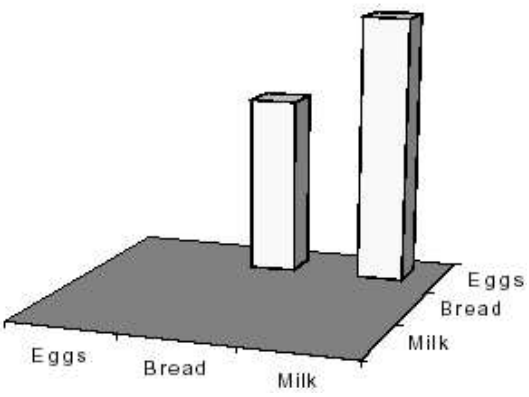


Figure 1. A 2-dimension matrix. Using this method to display many-to-one or many-to-many relationships is confusing. Here, we do not know if the representation means the rule $\{Bread,\ Milk\} \rightarrow \{Eggs\}$, or two rules: $\{Bread\} \rightarrow \{Eggs\}$ and $\{Milk\} \rightarrow \{Eggs\}$.

without scrolling. Furthermore, it is not easy to show multiple meta-data values such as support and confidence clearly. To illustrate, let us consider two rules: $\{a, b\} \rightarrow \{c\}$ and $\{a\} \rightarrow \{b, c\}$. For the first rule, we have node a and an edge to node b followed by node c. For the first rule, we can label the support and confidence on the edge connecting b and c. However, the problem arises when we include the second rule. If we now label the meta-data values on the edge between a and b, it becomes confusing for the same reason as the two dimension matrix. Creating a new set of edges, while solving the problem, turns the graph into an entangled Web.

Tables The table [3] is another technique for representing the textual results of data mining. Each row in the table represents a rule, and each rule is divided into various parts that is populated in the respective columns of the table. The advantage of this approach is the ability to sort the results by the column of interest (e.g., support). Hence, it is easy to identify rules with the highest confidence, lowest support or see the set of rules containing certain items in the antecedent or consequence. Beyond this, the limitation of the table is its close resemblance to the original raw textual form. As a result, it also lacked effective use of visual cues (e.g., colors and space) that can help enhance the organization of the rules discovered.

Grids An enhancement of the two dimension matrix is the grid [4]. Although a one-to-one relationship is displayed, it takes advantage of color cues to effectively present the results. Like the two dimension matrix, the axes represents

the antecedent and consequent items. In each cell of the grid, a color is assigned to indicate the confidence level while the tone of that color indicates the support. Hence, a brighter tone indicates a higher support and a brighter color indicates a higher confidence. Using color cues, the analyst can quickly obtain a summarized view of the results. For example, to find rules at high confidence with low support, the analyst simply identifies a particular color (indicating the high confidence) that has a lighter tone. Same as the two dimension matrix, the weakness lie in the lack of a practical way to identify the togetherness of items in a rule.

3. Proposed Improvements

The pre-condition to active mining is the ability to identify important rules quickly or easily. From the view point of visualization, the importance of the rules is never explicitly known and are at best expressed via the interestingness measures. Thus, the goal of visualization is to maximize the ease of identifying important rules. This lie in the use of visual cues such as color and space together with graphical metaphors to present both summarized views and incremental views. In particular, we focus on visualization techniques that are applicable in modern context where both the antecedent and consequent contain multiple items.

Noting the issues with the various visualization technique, we combined the two dimension matrix and grid to create a summarized view for fast identification of rules based on their support and confidence levels. We then improved the table view using a modern graphical metaphor to handle incremental detail views of all rules.

3.1. Enhanced Grid View

Figure 2 shows the enhanced grid view. The rows are ordered in increasing support and likewise, the columns are arranged in increasing confidence. Antecedent and consequent items are grouped and placed into the cells. Notice that this is a "flip" from the current approach. By doing this, we take advantage of the locality placement of a rule based on its support and confidence. Hence, looking for a strong rule (i.e., one with high support and confidence) means focusing on the bottom right corner of the grid where they are placed. Moving the mouse pointer to a particular cell in the grid pops a tooltip that show the actual rules placed into that particular cell. The enhanced grid view has the following merits:

- there is virtually no upper limit on the number of items that can exist in the antecedent or consequent, thus overcoming the many-to-many problem.
- the support and confidence of the association rules are clearly shown.

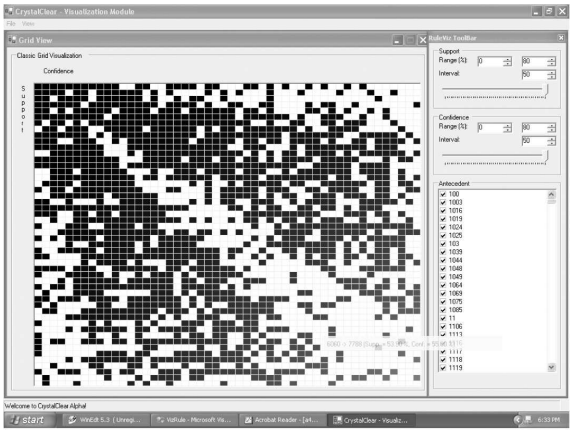


Figure 2. The enhanced grid view in our prototype: CrystalClear.

- the distribution of the association rules, as well as the items within the rules, can be analyzed simultaneously.
- the control panel (on the right in Figure 2) provides flexibility in selecting the rules to visualize by adjusting the support and confidence window.
- no screen swapping, animation, or complicated human interaction is required for analysis — only basic mouse movements.
- combining items in the antecedent or consequent to form a conceptual item for the purpose of overcoming the limits of one-to-one visualization in 2-dimension matrices is eliminated.

In the enhanced grid view, the analyst can control a number of parameters in visualization. As shown in the figure, the control panel on the right allows the definition of both the support and confidence window. By defining the window, the analyst can select the interested subset of rules. Within the control panel, the analyst can also determine the granularity of the grid. For example, if the support window is defined to select all rules that has a support in the range of 3% to 23%, and the granularity of the grid has been set to 10 cells, then each cell has an increment of 2% (same for confidence).

On top of that, better manageability is achieved with the ability to specify only items of interest. This is done by interactively checking items in the list on the control panel. As items are selected, rules satisfying all the constraints in the control panel are shown on the grid. Hence, the merit of the control panel lies in giving the analyst full control on what is to be visualized by the grid. In addition, the user interface in our prototype renders as soon as sufficient and valid information are in the control panel. This approach

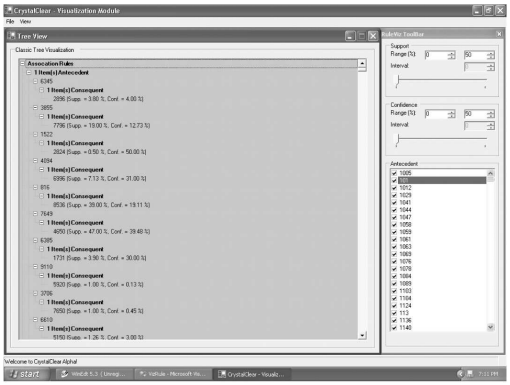


Figure 3. The tree view for organizing rules. The control panel has the same functionality discussed earlier.

gives real-time feedback to help analysts determine the best visualization setting.

3.2. Tree View

The primary objective of the grid is the ability to summarized the characteristics of a set of rules on a 2-dimension plane. In this section, we employ a well known GUI metaphor known as "TreeView". It allows many-to-many analysis where details of the rules are group by the similarity of their antecedent and consequent characteristics. It is an improvement over directed graphs and tables, and has the following advantages:

- similar to the 2-dimension grid, there is virtually no limit on the number of rules to be displayed.
- a hierarchical layout makes it easy to distinguish the nodes and edges from the graph and has similar organization capability as the table (i.e, sorting within a group based on highest confidence).
- meta information such as the number of rules with n-item consequence can be easily obtained since such information can be tagged to the meta-data nodes as shown in Figure 3.
- branches can be expanded or collapsed to control the rules to be displayed.

Up to this point, the two proposals are enhancements over the general techniques discussed in Section 2. These facilities would be sufficient if we are only interested in analyzing single set of rules. In reality, data conditions such as the removal of transactions, or the addition of a new item requires a re-run of the algorithm to maintain the relevance of the rules. Notice that each run of the algorithm generates a set of rules which is an evolution of its predecessor. Considering multiple sets of rules from the same but changing data sources, we realized that there are useful insights to be discovered. And Visualization is a good candidate for this task. We devote the next section for this discussion.

4. Visualizing Change for Active Data Mining

In the real world, data changes over time. This evolution of data is often a reflection of changes in the original observations. For example, the evolution of data in a set of customer transactions may reflect a gaining popularity of some goods or it may reflect the slowing demands of some products. In the same way, since knowledge are derived from snapshots of data, it is itself an evolving set of insights. Intuitively, if the changes in the data tells something about the trend of a business or some observation, then the way insights evolved over time is equally important.

Let $\mathcal{D}_i$ be the database at time t_i and $\mathcal{D}_j$ be the database evolved from $\mathcal{D}_i$ at time t_j such that $(i < j) \wedge (t_i < t_j)$. Let $r \in \mathcal{R}_i$ be the rule in the set of rules obtained from mining $\mathcal{D}_i$. In the context of visualization for association rules, insights can evolve in the following ways.

Case 1: A rule r that is interesting has become uninteresting because it fails to satisfy the support or confidence requirement at time t_j (i.e., $(r \in \mathcal{R}_i) \wedge (r \notin \mathcal{R}_j)$), and is thus removed from the new set of insights.

Case 2: A rule r is added into the new set of insights (i.e., $(r \notin \mathcal{D}_i) \wedge (r \in \mathcal{D}_j)$). This is the opposite of the first situation where a rule changes from uninteresting to interesting.

Case 3: A specific case of the above is when a new item (i.e., $(x \notin \mathcal{D}_i) \wedge (x \in \mathcal{D}_j)$) is added and new transactions due to x are created. This may generate new rules $\{r_1, r_2, \ldots, r_n\}$ such that $x \in r_k, 1 \leq k \leq n$.

Based on the three cases above, we extend our visualization capabilities to support the rendering of such observations using color cues and simple icons. We discuss this in the following sub-sections.

4.1. Active Mining Using Enhanced Grid View

The use of the grid view is to organize rules by their support and confidence across the visual space of the grid. Each rule is placed into one of the cell in the grid based on the rule's support and confidence, and are displayed via the tooltip when the mouse moves over a particular cell. The difference, in the case of active mining, is the use of two sets of rules instead of just one in the situation described in Section 3. In addition, colors are used to differentiate each of the three cases presented above.

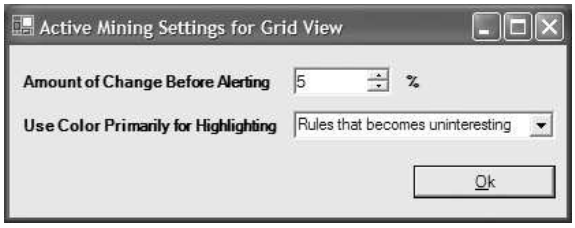


Figure 4. Control panel for for adjusting parameter values for active data mining.

For each case, we designate a specific color to represent a condition that has occurred in a particular cell. For example, if one of the rules in the cell fails to appear in the second set of rules, then the cell will be colored in red to indicate that one or more rules have become uninteresting. Likewise, when a new rule is added either due to the second or third case, a different color is used. Since within a cell, there can be multiple occurrences of the three cases, a way to identify each of the situation is needed.

The naive approach would be to display each color representing one of the cases above in the cell. One way to do this is to divide the cell into three parts, where each will show a color corresponding to one of the cases. The other alternative is to combine the color of each case using their RGB values to derive a new color representation. Both approach are inadequate. In the first case, dividing a cell into small sections create confusion about the size of the cell, making the visualization confusing. In the second case, deriving colors by combining existing ones makes color interpretation difficult as users have to remember the different color combinations. This defeats the original goal of visualization, which serves to focus rather then to confuse.

Our approach is to use icons together with colors to help enhance visualization. For each case above, we define a set of icons. For Case 1, we indicate with a '-' symbol. Likewise, for Case 2 and Case 3, we use '+' and '#' respectively. In addition, we also use '↑' and '↓' to indicate rules whose support and confidence have changed. Since a minor change may not be significant, we provided additional control parameters to allow user specify the threshold before highlighting the change. On the same control panel as shown in Figure 4, users can also specify how to use colors and icons. For example, if the user is interested mainly on rules that have become uninteresting (i.e., Case 1), then he or she can designate color as the primary indicator for change. For the other situations, icons that appear in the cell are used instead. This approach does not create visual distraction, maintains the original purpose of using color cues, and displays all changes that occurs in each cell. This is pictorially elaborated in Figure 5.

Figure 5. Using colors and icons in a cell for effective rendering of rule changes.

4.2. Active Mining Using Tree View

The concept discussed in the grid view is also applied to the tree view for incremental viewing of changes between two sets of rules. Special attention has been made to ensure that the colors and icons used are consistent with the grid view to avoid confusion in interpretation. Therefore, the same color code is used to highlight the nodes that indicate one of the three cases discussed. Icons are also used to show changes in the support and confidence, as well as to indicate summarized statistics of each case at their parent nodes.

The addition of summarized statistics at each parent node is useful in aiding the user in his or her decision of expanding a node. If a node's child has four rules falling in the Case 1 category, this will be shown using the icon '-' and a number that indicates that if the node is expanded (see Figure 6), there will be four rules that are previously interesting (in the first set of rules) and are now uninteresting (i.e., removed in the second set of rules).

Initially, the tree view organizes the rules by the number of items in the consequence and antecedence. To facilitate active mining of changes in rules, we further organize each rule by the type of changes it experience. This places the rules in one of the three cases or in the fourth case, where the rules remain unchanged in the previous and current sets of rules. This approach allows a more refine exploration and incremental view than what is shown in Figure 6.

5. Summary

We observed that the current state-of-the-art in association rule visualization is limited, and should be enhanced to meet the new challenges of rule visualization. We first propose two visualization metaphors (i.e., grid view and tree view) and then show how we can use visual and spatial cues to

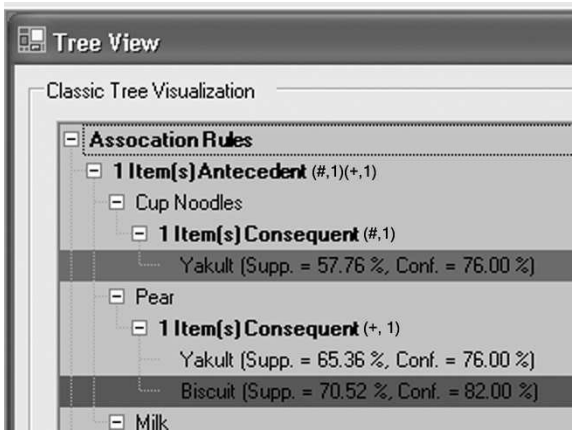


Figure 6. Active mining using the tree view.

enhance the summarization and organization of the rules discovered. We then consider the visualization of changes that has occurred from one set of rules to another, of which both sets of rules were obtained from the same database at two different instances. We argue that the visualization of change is important as it helps to identify various insights that cannot be detected by inspecting individual sets of rules. Such changes can detect sudden surge in the demand of a particular product or the buying behavior of customers (i.e., how bundling of items change).

Both the grid view and the tree view were prototyped in our application called *CrystalClear*. Currently, we are enhancing it to support the import of rules via the *Predictive Markup Modelling Language* (PMML) [19], an industry standard. PMML is supported in various packages such as *Microsoft SQL Server* and *PolyAnalyst*. Given the ability to read PMML, *CrystalClear* can be an add-on tool for many association rule mining packages. Taking this approach, we hope to position *CrystalClear* as a tool that will complement established data mining tools giving them additional capabilities to analyze data mining results.

The visualization presented in this paper is the first step to help expert analysis. In our future work, we are interested in using domain knowledge in visualization for better identification of useful patterns. This is motivated by our observation that some association rule algorithms use domain knowledge for pruning of irrelevant rules to improve human analysis [12, 20]. Hence, we believe visualization can also enhance the effectiveness of the analyst by incorporating domain knowledge.

References

[1] Advanced Visual Systems (AVS), OpenViz. *http://www.avs.com/software/soft_b/openviz/index.html*.

[2] DataLamp. Lanner, Graph Node. *http://www.datalamp.com*.

[3] Exclusive Ore Inc, XAffinity. *http://www.xore.com*.

[4] IBM Corporation. Quest, 3D Visualization of 2-items Rules. *http://www.almaden.ibm.com/cs/quest/demo/assoc*.

[5] Miner3D. Miner3D Excel. *http://www.miner3d.com/m3Dxl/features.html#builders*.

[6] VisualMine for Visual Analysis of Banking Activities. *http://www.visualmine.com/casestudy/casestudy.htm*.

[7] XGvis: A System for Multidimensional Scaling and Graph Layout in any Dimension. *http://www.research.att.com/areas/stat/xgobi*.

[8] R. Agrawal and R. Srikant. Fast Algorithm for Mining Association Rules. In *Proc. of the 20th Int. Conf. on Very Large Databases*, pages 487–499, Santiago, Chile, Aug. 1994.

[9] R. J. Bayardo and R. Agrawal. Mining the Most Interesting Rules. In *Proc. of the 5th Int. Conf. on Knowledge Discovery and Data Mining*, pages 145–154, San Diego, CA, USA, Aug. 1999.

[10] B. G. Becker. Volume Rendering for Relational Data. In *Proc. of Information Visualization*, Phoenix, Arizona, 1997.

[11] B. G. Becker. Visualizing Decision Table Classifiers. In *Proc. of Information Visualization*, Research Triangle Park, North Carolina, 1998.

[12] C. H. Cai, A. W. C. Fu, C. H. Cheng, and W. W. Kwong. Mining Association Rules with Weighted Items. In *Proc. of Int. Database Engineering and Applications Symp.*, Aug. 1998.

[13] D. Cheung, S. Lee, and B. Kao. A General Incremental Technique for Updating Discovered Association Rules. In *Proc. of the Int. Conf. on Database Systems for Advanced Applications*, Melbourne, Australia, Apr. 1997.

[14] D. W. Cheung, J. Han, V. T. Ng, and C. Y. Wong. Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique. In *Proc. of the 12th Int. Conf. on Data Engineering*, pages 106–114, New Orleans, Louisiana, USA, Feb. 1996.

[15] G. Dong and J. Li. Interestingness of Discovered Rules in Terms of Neighborhood-Based Unexpectedness. In *Proc. of the Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pages 72–86, Melbourne, Australia, Apr. 1998.

[16] J. Han and Y. Fu. Discovery of Multiple-Level Association Rules from Large Databases. In *Proc. of the 21th Int. Conf. on Very Large Databases*, Zurich, Swizerland, 1995.

[17] B. Hetzler, W. M. Harris, S. Havre, and P. Whitney. Visualizing the Full Spectrum of Document Relationships. In *Proc. of 5th Int. Conf. on Society for Knowledge Organization*, 1998.

[18] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns. In *Proc. of the 8th Int. Conf. on Knowledge Discovery and Data Mining*, Edmonton, Canada, 2002.

[19] D. Wettschereck and S. Muller. Exchanging Data Mining Models with the Predictive Modelling Markup Language. In *Proc. of Int. Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning*, Freiburg, 2001.

[20] O. R. Zaiane, J. Han, and H. Zhu. Mining Recurrent Items in Multimedia with Progressive Resolution Refinement. In *Proc. of Int. Conf. on Data Engineering*, San Diego, Mar. 2000.

Interactive Document Retrieval with Active Learning

Takashi ONODA, Hiroshi MURATA
Central Research Institute of Electric Power Industry
2-11-1 Iwato-kita, Komae, Tokyo 201-8511, JAPAN
E-mail: {onoda, murata}@criepi.denken.or.jp

Seiji YAMADA
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda, Tokyo 101-8430, JAPAN
E-mail: seiji@nii.ac.jp

Abstract

We investigate the following data mining problem from Information Retrieval: From a large data set of documents, we need to find those that bind to human interesting in as few iterations of human testing or checking as possible. In each iteration a comparatively small batch of documents is screened for binding the human interesting. We apply active learning techniques for selecting successive batches.

1 Introduction

As progression of the internet technology, accessible information by end users is explosively increasing. In this situation, we can now easily accsess a huge document database through the WWW. However it is hard for a user to retrieve relevant documents from which he/she can obtain useful information, and a lot of studies have been done in IR(Information Retrieval), especially document retrieval[19]. Active works for such document retrieval have been reported in TREC(Text Retrieval Conference)[16] for English documents, IREX(Information Retrieval and Extraction Exercise)[4] and NTCIR(NII-NACSIS Test Collection for IR System)[7] for Japanese documents.

In most frameworks for information retrieval, a vector space model in which a document is described with a high-dimensional vector is used[12]. An IR system using a vector space model computes the similarity between a query vector and document vectors by cosine of the two vectors and indicates a user a list of retrieved documents.

In general, since a user hardly describes a precise query in the first trial, interactive approach to modify the query vector by evaluation of the user on documents in a list of retrieved documents. This method is called *relevance feedback*[11] and used widely in IR systems. In this method, a user directly evaluates whether a document is relevant or no-relevant in a list of retrieved documents, and a system modifies the query vector using the user evaluation. A traditional way to modify a query vector is a simple learning rule to reduce the difference between the query vector and documents evaluated as relevant by a user.

Another approach has been proposed that classification learning with relevant and no-relevant document vectors as positive and negative examples for a target concept[8]. Some studies proposed SVM(Support Vector Machine) with excellent ability to classify examples into two classes is applied to classification learning of relevance feedback[15][3].

We propose a relevance feedback framework with SVM as *active learning*. In contrast that a conventional relevance feedback system indicates a user a list of the most relevant documents, our system provides a user a list of documents which are hard for SVM to classify them. This is a kind of active learning approach and we consider it promising for relevance feedback.

Okabe and Yamada[8] proposed a frame work in which relational learning to classification rules was applied to interactive document retrieval. Since the learned classification rules is described with symbolic representation, they are readable to our human and we can easily modify the rules directly using a sort of editor. However we consider SVM dealing with continuous values can do more precise classification than symbolic classification rules.

The relevance feedback is similar to what is termed active learning in that we try to maximize test performance using the smallest number of documents in the training set[15]. In active learning, we are interested in maximizing learning performance. Drucker et al.

applied SVM to the relevance feedback[3]. They are interested in maximizing the number of relevant documents which are displayed to users at each feedback iteration. However, we are interested in satisfying both the aim of active learning and the aim of increasing the number of relevant documents which are displayed to users at each feedback iteration. The detail of this difference will be described in the third section.

In the remaining parts of this paper, we explain a SVM algorithm in the second section briefly, and an active learning with SVM for the relevance feedback in the third section. In the fourth section, in order to evaluate the effectiveness of our approach, we made experiments using a TREC data set of Las Angels Times and discuss the experimental results. Eventually we conclude our work and discuss open problem in the fifth section.

2 Support Vector Machines

Formally, the Support Vector Machine (SVM) [17] like any other classification method aims to estimate a classification function $f : \mathcal{X} \to \{\pm 1\}$ using labeled training data from $\mathcal{X} \times \{\pm 1\}$. Moreover this function f should even classify unseen examples correctly.

In order to construct good classifiers by learning, two conditions have to be respected. First, the training data must be an unbiased sample from the same source (pdf) as the unseen test data. This concerns the experimental setup. Second, the size of the class of functions from which we choose our estimate f, the so-called capacity of the learning machine, has to be properly restricted according to statistical learning theory [17]. If the capacity is too small, complex discriminant functions cannot be approximated sufficiently well by any selectable function f in the chosen class of functions – the learning machine is too simple to learn well. On the other hand, if the capacity is too large, the learning machine bears the risk of overfitting.

In neural network training, overfitting is avoided by early stopping, regularization or asymptotic model selection [1, 6, 9, 10].

For SV learning machines that implement linear discriminant functions in feature spaces, the capacity limitation corresponds to finding a large margin separation between the two classes. The margin ϱ is the minimal distance of training points $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_i, y_i), \mathbf{x}_i \in \mathbf{R}, y_i \in \{\pm 1\}$ to the separation surface, i.e.

$$\varrho = \min_{i=1,\ldots,\ell} \rho(\mathbf{z}_i, f) \tag{1}$$

where $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ and

$$\rho(\mathbf{z}_i, f) = y_i f(\mathbf{x}_i), \tag{2}$$

and f is the linear discriminant function in some feature space

$$f(\mathbf{x}) = (\mathbf{w} \cdot \Phi(\mathbf{x})) + b = \sum_{i=1}^{\ell} \alpha_i y_i (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})) + b, \tag{3}$$

with $\mathbf{w}$ expressed as $\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \Phi(\mathbf{x}_i)$. The quantity Φ denotes the mapping from input space $\mathcal{X}$ by explicitly transforming the data into a feature space $\mathcal{F}$ using $\Phi : \mathcal{X} \to \mathcal{F}$. (see Figure 1). SVM can do so implicitly. In order to train and classify, all that SVMs use are dot products of pairs of data points $\Phi(\mathbf{x}), \Phi(\mathbf{x}_i) \in \mathcal{F}$ in feature space (cf. Eq. (3)). Thus, we need only to supply a so-called kernel function that can compute these dot products. A kernel function k allows to implicitly define the feature space (Mercer's Theorem, e.g. [2]) via

$$k(\mathbf{x}, \mathbf{x}_i) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i)). \tag{4}$$

By using different kernel functions, the SVM algorithm can construct a variety of learning machines, some of which coincide with classical architectures:

Polynomial classifiers of degree d:

$$k(\mathbf{x}, \mathbf{x}_i) = (\kappa \cdot (\mathbf{x} \cdot \mathbf{x}_i) + \Theta)^d \tag{5}$$

Neural networks(sigmoidal):

$$k(\mathbf{x}, \mathbf{x}_i) = \tanh(\kappa \cdot (\mathbf{x} \cdot \mathbf{x}_i) + \Theta) \tag{6}$$

Radial basis function classifiers:

$$k(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma}\right) \tag{7}$$

Note that there is no need to use or know the form of Φ, because the mapping is never performed explicitly The introduction of Φ in the explanation above was for purely didactical and not algorithmical purposes. Therefore, we can computationally afford to work in implicitly very large (e.g. 10^{10}- dimensional) feature spaces. SVM can avoid overfitting by controlling the capacity and maximizing the margin. Simultaneously, SVMs learn which of the features implied by the kernel k are distinctive for the two classes, i.e. instead of finding well-suited features by ourselves (which can often be difficult), we can use the SVM to select them from an extremely rich feature space.

With respect to good generalization, it is often profitable to misclassify some outlying training data points

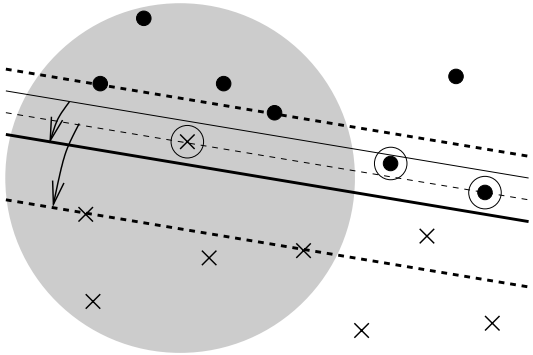


Figure 1: A binary classification toy problem: This problem is to separate black circles from crosses. The shaded region consists of training examples, the other regions of test data. The data can be separated with a margin indicated by the slim dashed lines, implicating the slim solid line as discriminate function. Misclassifying one training example(a circled white circle) leads to a considerable extension(arrows) of the margin(fat dashed and solid lines) and this fat solid line can classify two test examples(circled black circles) correctly.

in order to achieve a larger margin between the other training points (see Figure 1 for an example).

This soft-margin strategy can also learn non-separable data. The trade-off between margin size and number of misclassified training points is then controlled by the regularization parameter C (softness of the margin). The following quadratic program (QP) (see e.g. [17, 14]):

$$\begin{array}{lrll} \min & \|\mathbf{w}\|^2 + C\sum_{i=1}^{\ell}\xi_i & & \\ \text{s.t.} & \rho(\mathbf{z}_i, \boldsymbol{\alpha}) \geq 1 - \xi_i & \text{for all } 1 \leq i \leq \ell & (8) \\ & \xi_i, \alpha_i \geq 0 & \text{for all } 1 \leq i \leq \ell & \end{array}$$

leads to the SV soft-margin solution allowing for some errors.

3 Active learning with SVM in IR

In this section, we describe the information retrieval system using relevance feedback with SVM from an active learning point of view. In relevance feedback, the user has the option of labeling some of the top ranked documents according to whether they are relevant or not. The labeled documents along with the original request are then given to a supervised learning procedure to produce a new classifier. The new classifier is used to produce a new ranking, which retrievals more relevant documents at higher ranks than the original did. In relevant feedback method, the user have to judge the feedback documents. Hence, it is difficult to use a large number of user judged documents for supervised learning procedure because the user needs much effort to judged the documents. The SVMs have a great ability to discriminate even if the training data is small. Consequently, we propose to apply SVMs as the classifier in relevance feedback method. The retrieval steps of proposed method perform as follows:

Step 1: Preparation of documents for the first feedback
The conventional information retrieval system based on vector space model displays the top N ranked documents along with a request query to the user.

Step 2: Judgement of documents
The user then classifiers these documents into relevant or irrelevant. The relevant documents and the irrelevant documents are labeled.

Step 3: Determination of the optimal hyperplane
The optimal hyperplane is determined by using SVM which is learned by labeled documents.

Step 4: Discrimination all test collection and information retrieval
The SVM learned by previous step classifies the whole documents as relevant or not. The documents which are discriminated relevant and in the margin area of SVM are shown to user as the information retrieval results of the system. If the number of feedback iterations is more than m, then go to next step. Otherwise return to Step 2. The m is a maximal number of feedback iterations.

Step 5: Display of the final retrieved documents
The retrieved documents are ranked by the distance between the documents and the hyperplane which is the discriminant function determined by SVM. The retrieved documents are displayed based on this ranking.

The feature of our SVM-feedback is the selection of displayed documents to users in Step 4. Our proposed method select the documents which are discriminated relevant and in the margin area. In the reference [3], Drucker selects the higher ranked documents which are relevant and far from the discriminant function. This selection can not keep efficient learning from an active learning poin of view. In the reference [15], Tong selects the documents which are on or near the

discriminant function. This selection can make efficient learning. However, users feel stress of the selection because it is difficult to display the relevant documents by the selection. Our selection can be expected that the efficient learning can be kept and users do not need to feel stress.

4 Experiments

4.1 Experimental settings

We made experiments for evaluating the utility of our interactive document retrieval with active learning of SVM in §3. The document data set we used is a set of articles in the Los Angels Times(about 130 thousands articles, the average number of words in a article is 526) which is widely used in the document retrieval conference TREC[16]. This data set includes not only queries but also the relevant documents to each query. Thus we used the queries for experiments.

We used TFIDF[19], which is one of the most popular methods in IR to generate document feature vectors, and the concrete equations[13] are in the following.

$$
\begin{aligned}
w_t^d &= L * t * u \\
L &= \frac{1+\log(tf(t,d))}{1+log(\text{average of } tf(t,d)ind)} \quad (tf) \\
t &= \log(\frac{N+1}{df(t)}) \quad (idf) \\
u &= \frac{1}{0.8+0.2\frac{uniq(d)}{\text{average of } uniq(d)}} \quad (normalization)
\end{aligned}
$$

- w_t^d : Weight of a term t in a document d.
- $tf(t,d)$: Frequency of a term t in a document d.
- N : Total number of documents in a data set.
- $df(t)$: The number of documents including a term t.
- $uniq(d)$: The number of different terms in a document d.

The size N of retrieved results developed in **Step 1** in §3 was set as 20. The feedback iterations was 1, 2, 3 and 4. In order to investigate the influence of feedback iterations on accuracy of retrieval, we used plural feedback iterations.

In our experiments, we used the linear discriminant function for the classifier in SVM. The VSM of documents is high dimensional space. Therefore, in order to classify the labeled documents into relevant or irrelevant, we do not need to use the kernel trick and the the regularization parameter C(see §2). The VSM consists of TFIDF. Drucker et al. did not use TFIDF representation for SVM learning[3].

For comparison with our approach, two IR system were used. The first is a IR system that does not use feedback. The second is a IR system using traditional Rocchio-based releveance feedback[11] which is widely used in IR.

The Rocchio-based releveance feedback modifies a query vector Q_i by evaluation of a user using the following equation.

$$
Q_{i+1} = Q_i + \alpha \sum_{x \in R_r} x - \beta \sum_{x \in R_n} x, \qquad (9)
$$

where R_r is a set of documents which were evaluated as relevant by a user at the $i$the feedback, and R_n is a set of documents which were evaluated as no-relevant in the i feedback. α, β are weights for relevant, no-relevant documents respectively. we set $\alpha = 1.0$, $\beta = 0.5$ which are known adequate experimentally.

In general, retrieval accuracy significantly depends on the feedback iterations. Thus we changed feedback iterations for 1, 2, 3, 4 and investigated the accuracy for each iterations.

We utilized *precision* and *recall* for evaluating the three IR systems[5][18]. The following equations are used to compute precision and recall. Since a recall-precision curve is investigated to each query, we used the average recall-precision curve over all the queries as evaluation.

$$
\begin{aligned}
\text{Precision} &= \frac{\text{The No. of retrieved relevant doc.}}{\text{The No. of retrieved doc.}} \\
\text{recall} &= \frac{\text{The No. of retrieved relevant doc.}}{\text{The total No. of relevant doc.}}
\end{aligned}
$$

4.2 Experimental results

4.2.1 Comparing of recall-precision performance curve

In this section, we investigated the effectiveness of proposed method, when the user judged the top of twenty ranked documents at each feedback iteration. In the first iteration, twenty ranked documents were retrieved by VSM, which is represened by TFIDF.

Figure 2 show a recall-precision performance curve of SVM-based method, after four feedback iterations. For comparison, this figure also show the results of the conventional feedback method (i.e., Rocchio-based method) and VSM (i.e., without feedback). The thick solid line is the proposed method, the broken line is

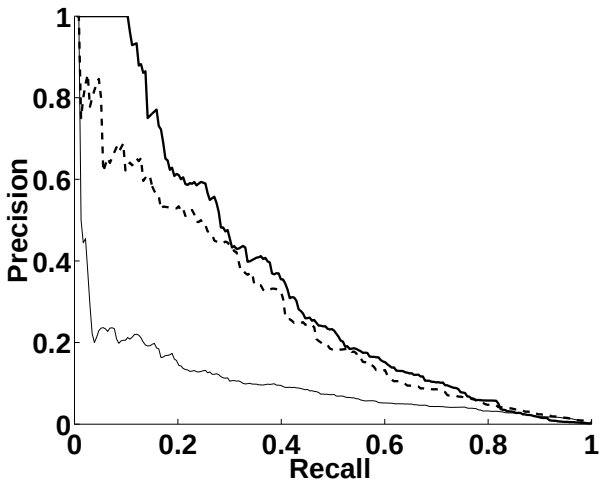


Figure 2: The effectiveness of SVM-feedback: The lines show recall-precision performance curve by using 20 feedback documents on the set of articles in the Los Angels after 4 feedback iterations. The wide solid line is proposed method, the broken line is conventional feedback method (i.e. Rocchio-based method), and the solid line is the VSM without feedback.

the conventional feedback method, and the thin solid line was the VSM without feedback.

This figure show that the retrieval effectiveness of both feedback methods, i.e., proposed and conventional feedback method, is improved compared with that of the non-feedback. In this result, we could confirm that the relevance feedback was useful technique for improving the performance of VSM.

Furthermore, this figure also show that the proposed feedback method improves the performance compared with conventional feedback method at all recall points. Consequently, we conclude that the SVMs are useful relevant feedback technique improving performance of VSM in this experiment.

4.2.2 Relationships between the performance and the number of feedback iterations

Here, we describe the relationships between the performances of proposed method and the number of feedback iterations. Table 1 gave the average precision result as a function of the number of feedback iterations. At each feedback iteration, the system displays twenty ranked relevant documents. We also show the average precision of Rocchio-based method for comparing to proposed method in table 1.

We can see from this table that the SVM feedback gives the higher performance in proportion to increase feedback iterations. On the other hand, the Rocchio-based feedback method degrades the retrieval performances nevertheless the number of feedback iterations increased from three to four. Hence, we can consider that the more feedback iterations, the better relevance documents they can obtain by using SVM-feedback method. Especially, the proposed method can improve the performance of conventional feedback method at each feedback iteration. We believe that the reason of these results was that the SVM can find a more suitable hyperplane for discriminating between relevant and irrelevant documents as increasing the feedback iterations. After all, we can believe that the proposed method can keep effective learning from active learning point of view.

Table 1: Average precision using SVM-feedback and Rocchio-feedback

No. of feedback iterations	Average precision	
	SVM	Rocchio
1	0.2625	0.2250
2	0.3500	0.2500
3	0.6125	0.2350
4	0.6375	0.2250

Furthermore, we compare the performances of proposed method to those of of Rocchiobased feedback method from an IR point of view. Table 2 shows the relationship between no. of feedback iterations and no. of actual relevant documents in 20 higher ranked relevant documents in a special case. In this special case, five documents were labeled as relevant documents in twenty documents at the first iteration. In almost case, one or two documents were labeled as relevant documents in twenty documents at the first iteration. We can see from this table that our SVM feedback can increase the number of actual relevant documents which are displayed to the user in proportion to increase feedback iterations. On the other hand, the Rocchio-based feedback method degrades the number of actual relevant documents nevertheless the number of feedback iterations increased from three to four. Hence, we can consider that the proposed method can give the suitable number of actual relevant documents to the user.

5 Conclusion

In this paper, we proposed the relevance feedback method with support vector machines (SVMs) for the information retrieval. Because the SVMs have an excellent ability to discriminate even if the training data is small, we applied the SVMs to relevance feedback

Table 2: In a special case, the relationship between no. of feedback iterations and no. of actual relevant documents in 20 higher ranked relevant documents.

No. of feedback iterations	No. of relevant documents	
	SVM	Rocchio
1	9	11
2	18	13
3	20	12
4	20	11

method. Experimental results on a set of articles in the Los Angels Times showed the proposed method gave a consistently better performance than the conventional feedback method. In our experiments we use TFIDF documents representation as VSM. Drucker et al. use binary documents representation and TF representation for estimating the performance of their proposed method. We plan to apply our proposed method to the binary representation and TF representation and compare our method with other SVM based methods(Drucker's method and Tong's method) experimentally.

In this paper, we proposed that the system should display the documents which are discriminated relevant and in the margin area of SVM at each feedback iteration. However, we do not discuss how the selection of documents influence both the effective learning and the performance of information retrieval theoretically. We would like to propose it as an open problem.

References

[1] C.M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[2] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *5th Annual ACM Workshop on COLT*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

[3] Harris Drucker, Behzad Shahrary, and David C. Gibbon. Relevance feedback using support vector machines. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 122–129, 2001.

[4] IREX. http://cs.nyu.edu/cs/projects/ proteus/irex/.

[5] D. Lewis. Evaluating text categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 312–318, 1991.

[6] N. Murata, S. Yoshizawa, and S. Amari. Network information criterion - determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks*, 5(6):865–872, 1994.

[7] NTCIR. http://www.rd.nacsis.ac.jp/˜ntcadm/.

[8] M. Okabe and S. Yamada. Interactive document retrieval with relational learning. In *Proceedings of the 16th ACM Symposium on Applied Computing*, pages 27–31, 2001.

[9] T. Onoda. Neural network information criterion for the optimal number of hidden units. In *Proc. ICNN'95*, volume 1, pages 275–280, 1995.

[10] J. Orr and K.-R. Müller, editors. *Neural Networks: Tricks of the Trade*. LNCS 1524, Springer Verlag, 1998.

[11] G. Salton, editor. *Relevance feedback in information retrieval*, pages 313–323. Englewood Cliffs, N.J.: Prentice Hall, 1971.

[12] G. Salton and J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[13] R.E. Schapire, Y. Singer, and A. Singhal. Boosting and rocchio applied to text filtering. In *Proceedings of the Twenty-First Annual International ACM SIGIR*, pages 215–223, 1998.

[14] B. Schölkopf, A. Smola, R. Williamson, and P.L. Bartlett. New support vector algorithms. Technical Report NC-TR-1998-031, Department of Computer Science, Royal Holloway, University of London, Egham, UK, 1998. *Neural Computation 2000*.

[15] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Journal of Machine Learning Research*, volume 2, pages 45–66, 2001.

[16] TREC Web page. http://trec.nist.gov/.

[17] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[18] I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, New York, 1994.

[19] R. B. Yates and B. R. Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

Chemical Data Mining Based on Structural Similarity

Yoshimasa Takahashi, Satoshi Fujishima, Kyoko Yokoe
Laboratory for Molecular Information Systems
Department of Knowledge-based Information Engineering, Toyohashi University of Technology,
Hibarigaoka 1-1, Tempaku-cho, Toyohashi 441-8580 JAPAN
Email: tala@mis.tutkie.tut.ac.jp

Abstract

This paper describes an approach to risk assessment of chemicals based on structural similarity. To validate an instance-based chemical risk report approach based on structural similarity, TFS-based similar structure searching was employed for identification of active molecular analogues. The TFS successfully identified structurally similar molecular analogues of our interest. The applicability of the TFS was validated also in discriminating active classes of pharmaceutical drugs. Dopamine antagonists of 1,227 that interact with different type of receptors (D1, D2, D3 and D4) were used for training an artificial neural network(ANN) with their TFS to classify the type of action. The ANN classified 87% of the drugs into their own classes correctly. Then, 79% of 137 chemicals were correctly predicted for a prediction set of 137 prepared in advance.

1. Introduction

We often say that "A is similar to B" or "C is similar to D in terms of xyz". "Similarity" is very important concept in solving problems in science. This is true in chemistry. The use of molecular similarity methods, especially structural similarity, is under active development in the area of drug design, for the selection of candidate analogs as new chemicals and for the estimation of molecular properties. Nevertheless, the concept of structural similarity is quite important for the further intelligent use of computers in the chemical field. The basic idea behind it is that structurally similar compounds are likely to possess similar molecular properties and similar biological activities. Most of the atomic groups defined in advance. However, the result of such a structural similarity analysis depends on the chosen set of substructures defined as the descriptors. In that case, an approach is required to process structural information in a more flexible way in order to allow somehow the automatic evaluation of the more ambiguous structural similarity; in other words, a method to examine the similarity of structures when they are regarded as whole entities.

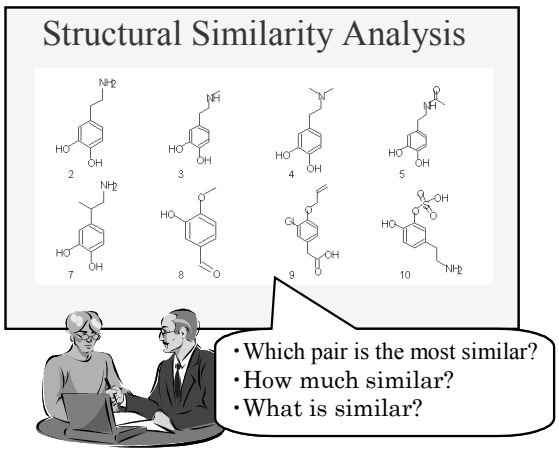


Figure 1. What is structural similarity analysis?

The aim of this research project is in establishing a basis of chemical data mining based on structural similarity without any set of substructures defined in advance. The authors proposed Topological Fragment Spectral (TFS) method as a tool for the description of the topological structure profile of a molecule. Here we investigate a more flexible way of structure handling based on TFS method. In this work, the applicability of the TFS method will be validated for similar structure-based risk reporting. In addition to this, discrimination of pharmacological activity classes of chemicals would be investigated using artificial neural network with the input signals of TFS descriptors.

2. Methods

TFS representation of chemical structure: In the present work, we investigate a numerical description

method of chemical structural information based on Topological Fragment Spectral (TFS) [1] and its application to chemical data mining based on structural similarity. The TFS is based on enumeration of all possible substructures from a chemical structure and numerical characterization of them. For a given structure represented as a chemical graph (hydrogen suppressed graph), all the possible subgraphs embedded in it are enumerated. Subsequently, every subgraph is characterized with a specific numerical quantity. To perform the characterization we have used two methods in the present study as follows: (i) the overall sum of degree of the nodes composing each subgraph. (ii) The overall sum of the mass numbers of the atoms (atomic groups) corresponding to the nodes of the subgraph. With the first method the chemical structure is represented by a simple graph thus the characterization of the structure depends only on the topology of the structural skeleton. For the second method, attached hydrogen atoms are taken into account as augmented atoms and are represented by weighting correspondingly their respective nodes in the graph. This is similar representation of mass spectra of chemicals. It is considered that the TFS is a function of chemical structure. An schematic flow of the TFS creation is shown in Fig. 2.

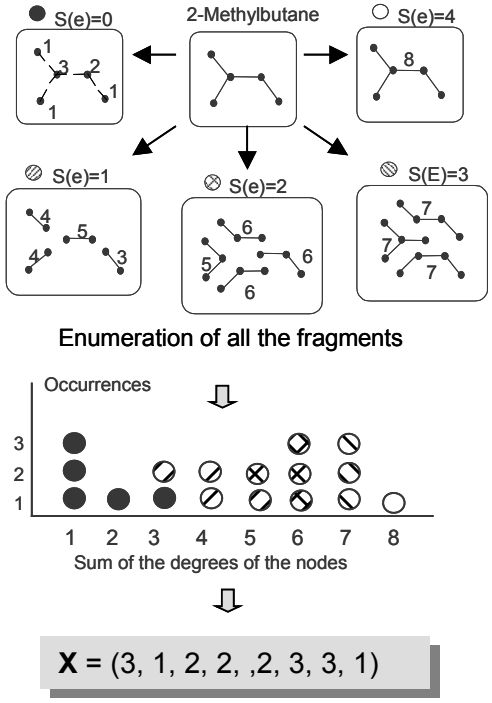


Figure 2. A schematic flow of TFS generation. S(e) is the number of edges (bonds) on the fragments to be generated.

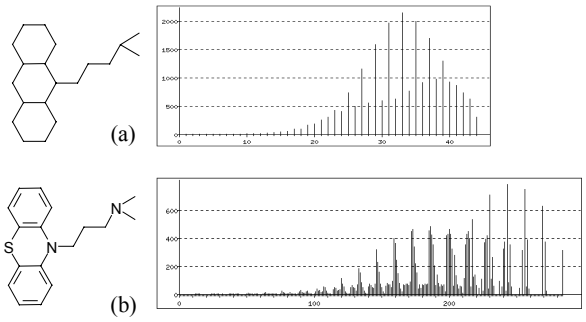


Figure 3. TFS of promazine generated by the different characterization methods.
(a) is characterized by the sum of degrees of nodes on the fragments. (b) is characterized by the sum of atomic mass numbers in fragments.

The TFS of promazine characterized by two different methods are shown in Fig. 3. The computational time required for the exhaustive enumeration of all possible substructures from a chemical structure is often very large especially for the molecules that involve highly fused rings. In addition to this, a large difference in the dimensionality between the fragment spectra to be compared may lead to the unexpected result. To avoid these problems an alternative approach based on the use of sub spectrum may be employed for such a similarity analysis, in which each spectrum can be described with structural fragments up to a specified size in the number of edges (bonds).

Quantitative evaluation of structural similarity based on the TFS: Obviously, the fragment spectrum obtained by these methods can be described as a kind of multidimensional pattern vector. Consequently, using this pattern representation of a spectrum it is possible to apply various quantitative measures for the evaluation of similarity. In the present work, Euclidean distance measure was used for evaluating the similarity.

$$D(X_i, X_j) = \sqrt{\sum (x_{ik} - x_{jk})^2} \qquad (1)$$

where, x_{ik} and x_{jk} are pattern vectors which represent the frequency value of peak k of fragment spectra of i-th molecule and j-th molecule respectively. $D(X_i, X_j)$ is the Euclidean distance between the patterns X_i and X_j. The different dimensionalities of the spectra to be compared are adjusted as follows,

If $X_i = (x_{i1}, x_{i2}, ..., x_{iq})$ and

$X_j = (x_{j1}, x_{j2}, ..., x_{jq}, x_{j(q+1)}, ..., x_{jp})$ (q<p),

then X_i is redefined as

$X_i = (x_{i1}, x_{i2}, ..., x_{iq}, x_{i(q+1)}, ..., x_{ip})$

here, $x_{i(q+1)} = x_{i(q+2)} = ... = x_{ip} = 0$.

This approach was fully computerized and used in the following structural similarity analysis and similar structure searching in a chemical database.

Data set: In this work we employed 1,337 dopamine antagonists that interact with four different types of receptors (D1, D2,D and D4). The data are a subset of MDDR[2] database. The data set was divided into two groups; training set and prediction set. The two include 1,227 compounds and 137 compounds respectively.

Neural network: Discrimination of pharmacological activity classes of chemicals was investigated using artificial neural network (ANN). Three-layer learning network with a complete connection among layers was used. The TFS was submitted to the ANN as input signals for the input neurons. The number of neurons in the input layer was 165, that is the same as the value of dimensionality of the TFS. The number of neurons in hidden layer was determined by trial and error. Training of the ANN was carried out by error back propagation method. All the neural network analyses were carried out using a computer program, NNQSAR, developed by the authors [3].

3. Results and Discussion

3.1 Classification of pharmacological activity using TFS/ANN:

The applicability of the TFS was validated in discriminating active classes of pharmaceutical drugs. Here, Dopamine antagonists of 1,227 that interact with different type of receptors (D1, D2, D3 and D4) were used for training an artificial neural network (ANN) with their TFS to classify the type of action. The ANN model with the obtained ANN model classified 89% of the drugs into their own classes correctly. Then, the trained ANN model was used for predicting class unknown compounds. For 137 separately prepared in advance the activity classes of 81% of the compounds were correctly predicted. All the results are summarized in Table 1.

Table 1. Results of neural network analysis

Class	Training		Prediction	
	No. of samples	Correct (%)	No. of samples	Correct (%)
All	1227	1087 (88.6)	137	111 (81.0)
D1	155	112 (72.3)	18	11 (61.1)
D2	356	312 (87.6)	39	27 (69.2)
D3	216	193 (89.4)	24	23 (95.8)
D4	500	470 (94.0)	56	50 (89.3)

In the comparison between the results it is shown that the results for D1 antagonists are poorer than other classes in both cases of training and prediction. It is considered that the ANN model wasn't learnt very much for the training set because the number of samples is relatively smaller than those of the other sets. The ANN had got the training for the compounds belonged to other three classes with their large number of samples. Beside, it is known that five compounds of the present set have antagonist activity for two or more classes. Taking account this matter, the total prediction rate was resulted in 84.7 %. These results show that the TFS is very powerful tool to describe structural information of chemicals and should be suitable as input signal to artificial neural network modeling for the classification and discrimination of pharmaceutical drug activities.

3.2 Risk report based on structural similarity:

To validate an instance-based chemical risk report approach based on structural similarity, TFS-based similar structure searching was employed for identification of active molecular analogues. The TFS database that consists of 1,227 drugs was prepared and used for the trial. The search trial with a query structure that has D1 antagonist activity resulted that all of first ten most similar compounds came from the same activity class (D1). Those chemical structures are shown in Fig. 4. The result shows that the TFS is powerful tool for similar structure-based risk report of chemicals.

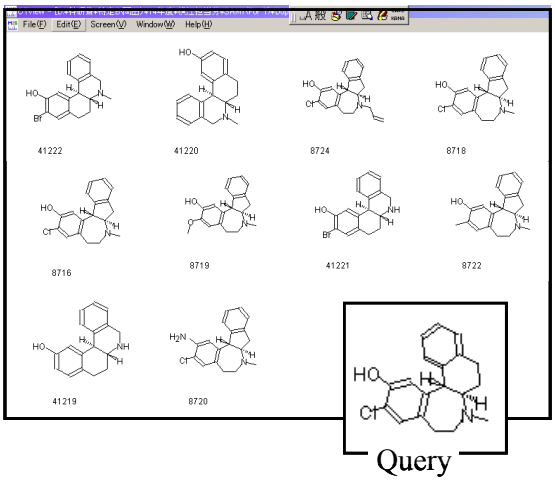


Figure 4. Similar structure searching by the TFS. Ten most similar structures obtained from 1,227 dopamine antagonists and the query are shown.

Next, we investigated the effect of noise compounds at the similar structure searching, because only dopamine antagonists were employed for this trial and their chemical structures might be limited into the small world

in terms of structural diversity. We took other 10,000 compounds from MDDR, and generated the TFS for all the compounds. These data were added to the former TFS database. The same query was employed for the comparison between the results for the former database and the extended database. However, the computational experiment gave us the completely same result of the first ten most similar structures even for the extended database that involves 11,227 compounds; 1,227 dopamine antagonists and 10,000 other drugs. It was concluded that the TFS-based similarity searching gave us successful result for this purpose.

3.3 Visualization of TFS similarity space:

A desktop software tool, MolSpace [4], was used for visualizing massive molecular data space. MolSpace can project a set of massive multivariate data (e.g. TFS data) onto a visual space (2D or 3D space) by means of principal component analysis. MolSpace allows users not only to draw a scatter diagram of the data but also to display their 2D or 3D molecular structures as the objects in the space. With a probe (a molecular object) the user can navigate vast data spaces, thus facilitating understanding of the data structure. In addition, partial space searching is also available that is based on similarity searching techniques described above. It is possible to interrogate a 3D structure of a chemical compound that corresponds to each object on the space in real time. An example for the current work is shown in the below.

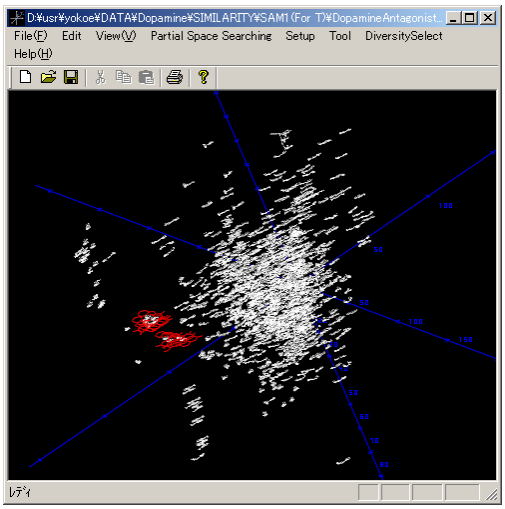


Figure 5. Visualization the TFS chemical data space reduced into 3D-space by PCA mapping.

Figure 5 shows the TFS similarity space of the training set of 1227 drugs. The objects colored by red are first ten most similar compounds for the query shown in Fig.4. We can see that these similar compounds are located at near space in each other.

4. Future work

Because many instances are required for predictive risk assessment and risk report, more large set of real data should be used in further work. For the purpose, a large size of TFS database of 120,000 pharmaceutical drugs is under preparation, and it would be used to improve the classification model and to find the similar molecules using similar structure searching. Additional system that can be used for identification and interpretation of the TFS peaks of our interests will be also required.

This work was supported by Grant-In-Aid for Scientific Research on Priority Areas(B) 13131210.

5. References

[1] Y. Takahashi, H. Ohoka, and Y. Ishiyama, Structural Similarity Analysis Based on Topological Fragment Spectra, *In "Advances in Molecular Similarity"*, **2**, (Eds. R. Carbo & P. Mezey), JAI Press, Greenwich, CT, 1998, pp.93-104 (1998)

[2] MDL Drug Data Report, MDL, ver 2001.1, (2001).

[3] H. Ando and Y. Takahashi, Artificial Neural Network Tool (NNQSAR) for Structure-Activity Studies, *Proceedings of the 24th Symposium on Chemical Information Sciences*, 2000, pp.117-118.

[4] Y. Takahashi, M. Konji, S. Fujishima, MolSpace: A Computer Desktop Tool for Visualization of Massive Molecular Data*, J. Mol. Graph. Model*., in press.

Mining Hepatitis Data Set Using Information Gathered from Biomedical Literature

TuanNam Tran
Dept. of Computer Science
Tokyo Inst. of Technology
tt-nam@nm.cs.titech.ac.jp

Ryutaro Ichise
Knowledge Systems Research
National Inst. of Informatics
ichise@nii.ac.jp

Masayuki Numao
Dept. of Computer Science
Tokyo Inst. of Technology
numao@cs.titech.ac.jp

Abstract

This paper presents a novel technique for mining medical data taking into account the importance of each attribute occurred in the biomedical literature. Our approach attempts to combine active information gathering and user-centered mining stages of the active mining framework. Our method, which is based on C4.5rules, considers the external weight of each attribute which can be calculated by means of the number of corresponding documents found in the biomedical literature. The experimental result on hepatitis data set shows that the proposed method may be useful in terms of generating interesting rules from the viewpoint of domain experts.

1. Introduction

Data mining is the technique that aims at extracting useful knowledge from huge amount of data [3]. There have been many data mining systems based on rule discovery, and it has been known that domain experts play an essential role in discovering novel knowledge [12]. Many of those systems generate rules based on factors such as support or accuracy, however as indicated in [19], applying those systems manually without background knowledge on the given data often generates common but not interesting rules from the viewpoint of domain experts. Conversely, rules which are considered to contribute to the mining process tend to have not so hight support and confidence.

This paper extends our previous work on mining meningoencephalitis diagnosis data set [18], by combining C4.5rules [13], a standard mining algorithm, and *information gathering* obtained from biomedical literature considering the number of documents related to each attribute of the given data. We propose a new method which focuses on both data mining techniques as well as other techniques concerning *information gathering* such as information retrieval and text mining. In this paper, our proposed method is evaluated on hepatitis data set used in the Active Mining project as well as at some workshops such as ECML/PKDD2002 Workshop on Discovery Challenge [1].

The remainder of this paper is organized as follows. Section 2 describes our algorithm in detail. The hepatitis data set and its pre-processing will be presented in Section 3. Section 4 reports some experimental results obtained from the hepatitis data set. Section 5 demonstrates the discussions related to the proposed method. Some related work will be described in Section 6, and finally Section 7 presents our conclusion.

2. Finding the best attribute using MEDLINE

As mentioned above, a large number of conventional mining methods pay attention only to the given data itself, not considering other external factors such as scientific sources of literature or domain experts' background knowledge concerning the data. Although combination of conventional data mining methods and domain experts' background knowledge is a natural idea and may improve the mining results, this approach requires more time and cost for the domain expert themselves, and in fact, cannot be implemented easily. For this reason, this work aims to combine information gathering, biomedical literature retrieval techniques with standard data mining techniques in order to generate rules which are interesting to the medical experts, although their support may be lower.

Our approach is to modify the gain of C4.5 by using external weights, which are calculated by using MEDLINE, a premier source of bibliographic coverage of biomedical literature produced by the National Library of Medicine. PubMed [8] is an on-line MEDLINE search system provided by the National Library of Medicine (NCBI). To date, it contains approximately 12 million citations back to 1966, and about 400,000 new citations is added to MEDLINE each year. The citations are taken from over 3900 journals and

about 80% of them contain abstracts.

Differing from conventional mining methods (for mining medical data), our approach assumes that the information of MEDLINE documents concerning the attributes of the given data is also useful for mining. In detail, we hypothesize that the larger the number of MEDLINE documents concerning an attribute, the more its external weight is. Our proposed algorithm inherits C4.5's gain and gain ratio calculations and the external weight of attributes calculated by querying MEDLINE.

The core of a decision tree algorithm is to repeat the process of selecting the attribute with highest information ratio. The characteristic of our method is to consider "weighting"by external information when calculating information ratio for each attribute. If the importance of an attribute (i.e. the number of literature documents from previous biomedical research) occurred in the given data can be calculated, the importance of that attribute can be calculated easily.

Suppose T is a set of training examples of a decision tree consisting of attributes $A_1, A_2, \ldots, A_m$, and $freq(C_i, S)$ is the the number of cases in a set of examples S that belong to class C_i. The $entropy$ of T is defined as:

$$info(T) = -\sum_{j=1}^{k} \frac{freq(C_j, T)}{|T|} \times log_2 \left(\frac{freq(C_j, T)}{|T|} \right)$$

where $freq(C_j, T)$ stands for the number of cases in T that belong to class C_j.

Suppose T has been partitioned in accordance with the n outcomes $T_1, T_2, \ldots, T_n$ of a test X corresponding to the attribute A_j. Then, according to [13], $gain$ and $gain\ ratio$ for the attribute A_j at the given stage in the construction of the decision tree can be calculated as follows:

$$info_j(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times info(T_i) \quad (1)$$

$$gain_j = info(T) - info_j(T) \quad (2)$$

$$split\ info_j = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \log_2(\frac{|T_i|}{|T|}) \quad (3)$$

$$gain\ ratio_j = gain_j / split\ info_j \quad (4)$$

The *external weight* $\omega(A_j)$ of an attribute A_j using the biomedical literature information is defined, and $gain$, $gain\ ratio$ are modified as follows:

$$\omega_j = \frac{F(|A_j|)}{\sum_{i=1}^{m} F(|A_i|)} \quad (5)$$

$$gain_j^{'} = gain_j \times \omega_j \quad (6)$$

$$gain\ ratio_j^{'} = gain\ ratio_j \times \omega_j \quad (7)$$

Here, $|A_i|$ is the number of MEDLINE documents found related on the given data and the attribute A_i.

We have currently defined two types of $F(x)$ as follows:

$$F_1(x) = x \quad (8)$$

$$F_2(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ \lfloor (log_k(x) + 1) \rfloor & \text{if } x > 0 \end{cases} \quad (9)$$

where $k > 0$

C4.5 selects the attribute that maximizes the information gain ratio ($gain\ ratio$), which is a function of the information gain, and we modified C4.5 so that it selects the attribute that maximizes $gain\ ratio'$.

Our algorithm is similar with some other algorithms such as EG2 [11], CS-ID3 [16], IDX [10] in terms of modifying the information gain for selection of attributes, however it should be noted that they aim to minimize the costs of tests, while our purpose is to find interesting patterns for the domain experts.

3. Hepatitis data set

3.1. Introduction to the hepatitis data set

Hepatitis virus B and C are major ones among chronic viral hepatitis. There are infective diseases which the tissue of the liver is chronically inflamed by the continuous infection of the hepatitis virus. Since chronic viral hepatitis has a potential risk of developing liver cirrhosis and hepatoma, the task of mining hepatitis data to extract novel knowledge is considered as a desirable way to further investigate about hepatitis.

The hepatitis data set contains administrative information as well as long time-series data of laboratory examinations of 771 patients with hepatitis B and C who took biopsy in the period 1982-2001 at Chiba University hospital, Japan. The data are organized in six tables: basic information about the patients (*pt*), results of biopsy (*bio*), information on interferon therapy (*ifn*), results of out-hospital examinations (*olab*), results of in-hospital examinations (*ilab*) and information about measurements in in-hospital examinations (*labn*). The first three tables include administrative information and the last three tables include examination data of blood test and urinalysis. The in-hospital data contain the results of 230 examinations that were able to conduct inside the hospital and the out-hospital data contain the results of 753 examinations which were performed using special equipments from other facilities. As a result, the examination data contain 983 types of examination.

The ultimate goals of mining hepatitis are to discover the relationships between the stage of liver fibrosis and the onset of hepatocarcinoma, the differences in temporal patterns between hepatitis B and C, the relationships between hematological status and time to the onset of hepatocarcinoma,

evaluating whether laboratory examinations can be used to estimate the stage of liver fibrosis, whether the interferon therapy is effective or not and if GOT and GPT can be used to measure the inflammation speed [21].

3.2. Pre-processing

In general, the purpose of the pre-processing stage is to generate a single table from the given six tables after conducting data cleaning, attribute selection, attribute generation using MID, a primary key given for each patient, and the examination data of the patients.

- For *ilab* table:
  With the guidance of medical experts, we have classified all examinations into two groups as follows:

  1. *essential examinations*: This group can be classified into two sub-groups. The *short-term-change* category containing GOT, GPT, TTT, ZTT may change considerably for a relatively short time, and the *long-term-change* category containing T-CHO, CHE, ALB, TP, T-BIL, D-BIL, I-BIL, ammonia, ICG-15 which only change considerably for a relatively long time.
  2. *others*: other examinations.

  All of in-hospital examinations can be discretized into $+$ (abnormal value) or $-$ (normal value) according to the corresponding information from the *labn* table. Although discretization makes the generated rules easier to comprehend, we have currently discretized only the *others* group, and used the continuous values for the *essential examinations* group, since the discretization of all examinations may lead to the loss of information such as *trend* in time-series data. We have also constructed new attributes called "interferon dosing state" which may take the value of *before*, *during*, *after* using the *ifn* table. In addition, a new attribute which can take the value of *response*, *partial response*, *aggravation* or *no change* was constructed using the judging standard of interferon effect created by the Hepatitis Research Group on hard-to-cure disease of the Ministry of Health and Welfare, Japan.
- For *olab* table:
  Although this table contains more than 75% of the total examinations in which many of them are measured only a few times. According to the advice of domain experts, we only focus on whether the results are positive(abnormal, $+$) or negative (normal, $-$), not considering their numeric values. For those examinations which cannot be decided as positive or negative, we use the $+-$. In addition, we also extracted blood type from *olab* as a new attribute.
- For *bio* table:
  Since the original English version of given data do not contain biopsy results as well as facility information, we extracted and merged into *bio* table this information from the Japanese version of the given data. From the biopsy results we constructed new attributes such as LC, CAH, CPH which can take the value of "y" or "n". For instance, "LC = n" means that it is not a liver cirrhosis state.

Using *pt* table, we have calculated and constructed a new attribute Age at every examination date for every table mentioned above. In order to use decision tree systems, we have integrated those three tables into a single table according to MID and examination date. The key point is that since the examination date for each MID of *ilab*, *olab* and *bio* tables may be different and the average number of records per MID of *ilab* table is considerably larger than those of *olab* and *bio* tables, we have decided to change each examination date of *olab* and *bio* tables to the nearest examination date of the same patient in *ilab* table. Attributes which occur in both *ilab* and *olab* tables (CRP, HBC-AB, HBE-AB, HBE-AG, HBS-AB) are then merged to eliminate duplicated attributes.

4. Experiments

We have currently considered the following problems:

- Discover knowledge concerning the stage of liver fibrosis using laboratory examinations
- Discover knowledge which indicates whether the interferon therapy is effective or not.
- Discover knowledge which distinguishes hepatitis B and C.

Figure 1, 2 and 3 show some obtained rules with a relatively high accuracy as well as their respective evaluations of the domain experts.

4.1. Discover patterns concerning liver fibrosis

Figure 1 shows some rules concerning the liver fibrosis. The figures in [ ] respectively indicate the number of patients satisfying the antecedent of the rule and number of patients not satisfying the rule itself. The symbols "*" and "**" indicate that the corresponding rule is obtained by C4.5rules or by our proposed algorithm, respectively. The italic symbols F_1 and F_2 mean that the corresponding rule is found by function F_1 or F_2. For instance, the first part of rule 1 in Figure 1 implies that this rule can be obtained by both of C4.5rules and our proposed method, while the second part "if $hepatitstype = B$ then $BIOPSYFibrosis > 1$" was generated only by our proposed method (using function F_1).

1. IF (hepatitis type = C)
THEN BIOPSY Fibrosis = 1 [297/132] *,** (F_1)

IF (hepatitis type = B)
THEN BIOPSY Fibrosis $>$ 1 [206/79] ** (F_1)

Evaluation: From the viewpoint of biopsy fibrosis, this rule means that the number of patients with hepatitis B are high, and as a result not contradict to the rules 1, 2 shown in Figure 3.

2. IF (CRP = $+-$) AND (GPT $\leq$ 67) AND (hepatitis type = C)
THEN BIOPSY Fibrosis $>$ 2 [19/6] *

Evaluation: This rule may be some interesting because it shows the relationship between CRP and hepatitis which has not yet been pointed out so far.

3. IF (GPT $>$ 33) AND (hepatitis type = B) AND (TP $>$ 6.8) AND (LC = n)
THEN BIOPSY Fibrosis = 2 [75/43] ** (F_2)

Evaluation: It is difficult to judge.

4. IF (RA = $-$)
THEN BIOPSY Fibrosis $<$ 3 [19/1] *,** (F_2)

IF (senketsu (occult blood) = +)
THEN BIOPSY Fibrosis $<$ 3 [14/1] *,** (F_2)

IF (IFN effect = partial response)
THEN BIOPSY Fibrosis $<$ 3 [35/5] ** (F_1)

Evaluation: It is considered that there are few clinical meanings.

Figure 1. Some obtained rules concerning liver fibrosis

4.2. Discover relationships concerning interferon treatment effect

The fact that interferon treatment brings about a cure for hepatitis C has been already known, however it does not mean that interferon treatment is effective for every patient with hepatitis C. For this reason, it is important to discover knowledge concerning the effect of the interferon treatment. Figure 2 shows some rules concerning the effect of the interferon treatment obtained by our system. In this figure, rule 1, 2 and 3 forecast the effect of the interferon treatment. It can be seen from these rules that the in-hospital examinations may be used for predicting the effects of interferon treatment. It is by chance that no rule with the proposed method is shown, since we have currently chosen subjectively only rules with a relatively high accuracy.

4.3. Discover patterns which distinguish hepatitis virus type B and hepatitis virus type C

1. IF (IFN dosing state= before) AND (I-BIL $\leq$ 0.3) AND (TP $\leq$ 7.6) AND (BIOPSY Fibrosis $\leq$ 3)
THEN IFN effect = aggravation [10/4] *

Evaluation: The IF portion is almost meaningless.

2. IF (IFN dosing state = before) AND (T-BIL $>$ 1.1) AND (TP $\leq$ 7.6)
THEN IFN effect = no change [11/2] *

Evaluation: T-BIL$>$1.1 means that T-BIL is greater than the upper normal value, and this rule is interesting, since it has not been pointed out so far that in that condition interferon may not be effective easily.

3. IF (IFN dosing state = before) AND (T-BA = +)
THEN IFN effect = no change [8/4] *

Evaluation: It is interesting that when T-BA is abnormal, the effect of interferon is unchanged. It may be discovery if the hypothesis that interferon is not effective when there is an excretion obstacle of liver is hold.

Figure 2. Some rules concerning interferon treatment effect

Figure 3 shows some rules distinguishing hepatitis B and C. Rule 1 and 2 mean that it can judge the type of hepatitis whether B or C using the examination CHE. Rule 4 and rule 5 mean that when the biopsy activity is 0 or FALSE, the hepatitis type will be C. This indicates that biopsy activity is an essential factor for judging whether the hepatitis type is B or C.

5. Discussion

The way of choosing best attributes using the number of MEDLINE's corresponding documents makes the attributes with high external weighting values being taken priority over the low ones. That is, the occurring probability of those attributes paid attention to in the literature will become higher. Table 1 shows an example in which CHE is chosen either using the proposed function F_2 (with k = 10) or without using the proposed method, but HBE-AG is chosen when using the function F_1.

The merit of using information gathered from biomedical literature is that, the users are able to know the "importance" of each attribute without any assistance of the domain experts. Moreover, the generated rules of our algorithm are able to reflect the state-of-the-art research in biomedical literature, since the number of documents related to an attribute, and as a result its external weight changes with time. One more merit of our system is that it is easy and flexible to update the weighting scheme. That is, we can increase the weight of the attributes that were highly evaluated by the domain experts, and by repeating the mining process we may obtain

1. IF (CHE $>$ 12.58)
THEN hepatitis type = C [259/35] *,** (F_2)

IF (CHE $\leq$ 12.58)
THEN hepatitis type = B [117/15] ** (F_2)

Evaluation: In general, hepatitis B shows a lower value of CHE compared to hepatitis C, and it is possible to say that hepatitis B is more progressive than hepatitis C.

2. IF (U-BIL = $-$)
THEN hepatitis type = C [152/21] *,** (F_2)

IF (LDH = +) AND (ZTT $\leq$ 13.8) AND (BIOPSY Fibrosis $>$ 1) AND (BIOPSY Activity $>$ 0)
THEN hepatitis type = B [61/2] ** (F_2)

Evaluation: It is difficult to judge.

3. IF (BIOPSY Activity $\leq$ 0)
THEN hepatitis type = C [48/0] ** (F_1)

Evaluation: It can be explained that there is no person whose biopsy activity of hepatitis B is 0, although this is medically hard to judge.

4. IF (CHE $\leq$ 85) AND (GPT $>$ 71) AND (GPT $\leq$ 92) AND (BIOPSY Fibrosis $\leq$ 1)
THEN hepatitis type = B [10/0] ** (F_1)

Evaluation: It is difficult to judge.

Figure 3. Some rules for distinguishing hepatitis virus type B and type C

new interesting rules.

In general we have obtained some rules which were evaluated as "interesting" by domain experts. The domain experts also gave us valuable comments on the idea of weighting attributes using MEDLINE search as necessary and at the same time, it may generate knowledge which has already been known since our approach focuses on those attributes which are cited frequently in the literature, and it is impossible to conduct mining effectively without taking into account the past knowledge accumulated in the literature.

As for future work, we are planning to increase the number of weighting functions, for instance those that were used in the previous work on cost-sensitive learning (mentioned in Section 2). The idea of *temporal abstraction* mentioned in Section 6 is also interesting, since it can decrease the number of examples in the given table as well as showing the trend of examinations in time-series data. One of the characteristics of the hepatitis data set is its unbalance for each patient, i.e. in the outcome table of pre-processing, there may have a quite large number of cases for a patient, while only a quite small number of cases for another one. Our current algorithm as well as the standard C4.5 do not consider this situation of time-series data, thus may generate rules which cover a large number of cases, but only satisfy

Table 1. An example shows the external weights of CHE and HBE-AG for the problem mentioned in Section 4.3

Attribute	Function	
	F_1	F_2 (k=10)
CHE	0.000	**0.015**
HBE-AG	**0.096**	0.030

a relatively few patients. We may address this problem by modifying the heuristic functions taking into account the MIDs. It is also interesting to reduce the number of documents related to an attribute by focusing on only those contain *interesting knowledge* using a supervised machine learning approach and utilizing more semantic information found in the documents.

6. Related work

Blum and Langley [2] have reviewed the problem of *feature selection*, i.e. selecting the most relevant features in representing the data. In their paper, some explicit feature selection approaches characterized as 'embedded','filter' or 'wrapper' are compared to to those based on weighting schemes. With respect to *feature weighting* methods, there have been well-known algorithms such as *perceptron*, *WINNOW* and *backpropagation*. In decision tree induction, some work has been conducted on *cost-sensitive classification* which consider either the costs of tests (features, measurements) or the costs of classification errors. For instance, there are several machine learning algorithms that consider the costs of tests such as EG2 [11], CS-ID3 [16] and IDX [10]. Some other studies consider the weighting scheme for instances such as the boosting algorithm [4] or attempt to adapt the boosting algorithm for cost-sensitive classification [17]. Turney [20] introduced a method which uses a genetic algorithm with the fitness function is the average cost of classification when using the decision tree, including both the costs of tests and the costs of classification errors.

On the other hand, there have been a number of studies with respect to the task of evaluating and comparing various data mining methods using medical data [15], [19], [1]. There have also been some other approaches for mining hepatitis data set. Ho et al. [7] used their visual data mining system D2MS and *temporal abstraction* [14], a knowledge-based framework for the creation of abstract, interval-based concepts from time-stamped medical data. Matsuda et al. [9] transformed the given data into the graph-structure one and applying a graph-based induction approach for extracting typical patterns from graph data. Another approach focused on the sequential pattern analysis is described in

[5]. Hirano et al. presented a method for analyzing time-series based on the phase-constraint multi-scale matching and rough clustering [6]. All of the above approaches differ from our method, since they have not considered external sources concerning the given data.

7. Conclusions

We have proposed a novel mining method for mining medical data which combines biomedical search information and C4.5rules, a well-known mining algorithm. The core of our method is an algorithm in which we have used some heuristic functions considering the external weight of each attribute calculated by the number of literature documents found for the corresponding attribute. We have tested the proposed method on the hepatitis data set, and showed that our proposed method may be useful in terms of generating interesting rules to the domain experts.

Acknowledgements

This research is supported by the grant-in-aid for scientific research on priority area "Active Mining" from the Japanese Ministry of Education, Science Sports and Culture. The authors would like to thank Dr.Katsuhiko Takabayashi, Dr.Hideto Yokoi of Chiba University Hospital as well as anonymous reviewers for their useful comments on our work.

References

[1] P. Berka, J. Rauch, and S. Tsumoto, editors. *Proc. of ECML/PKDD2002 Workshop on Discovery Challenge*, 2002. http://lisp.vse.cz/challenge/ecmlpkdd2002/.

[2] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1–2):245–271, 1997.

[3] U. M. Fayyad. *Advances in Knowledge Discovery and Data Mining*. AAAI Press, 1996.

[4] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proc. of the 13th International Conference on Machine Learning (ICML)*, pages 148–156, 1996.

[5] H. Hatazawa, Y. Sato, and Y. Yamaguchi. Rule discovery based on sequential pattern analysis and mining. in the case study of chronic hepatitis datasets. Technical Report SIG-KBS-A201, Japanese Society for Artificial Intelligence, 2002. In Japanese.

[6] S. Hirano, X. Sun, and S. Tsumoto. Analysis of time-series medical data based on similarity of convex/concave structure of sequences. Technical Report SIG-KBS-A201, Japanese Society for Artificial Intelligence, 2002. In Japanese.

[7] T. B. Ho, D. D. Nguyen, S. Kawasaki, and T. D. Nguyen. Extracting knowledge from hepatitis data with temporal abstraction. In *Proc. of ECML/PKDD2002 Workshop on Discovery Challenge*, 2002.

[8] PubMed homepage. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi.

[9] T. Matsuda, T. Yoshida, H. Motoda, and T. Washio. Knowledge discovery from hepatitis data based on graph structure. Technical Report SIG-KBS-A201, Japanese Society for Artificial Intelligence, 2002. In Japanese.

[10] S. W. Norton. Generating better decision trees. In *Proc. of the 11th International Join Conference on Artificial Intelligence (IJCAI)*, pages 800–805, 1989.

[11] M. Núñez. The use of background knowledge in decision tree induction. *Machine Learning*, 6(3):231–250, 1991.

[12] G. Piatetsky-Shapiro and W. J. Frawley, editors. *Knowledge Discovery in Databases*. AAAI Press, 1991.

[13] J. R. Quinlan. *C4. 5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[14] Y. Shahar and M. A. Musen. Knowledge-based temporal abstraction in clinical domains. *Artificial Intelligence in Medicine*, 8:267–298, 1996

[15] E. Suzuki, editor. *Proc. of PAKDD2000 Workshop on KDD Challenge*, 2000. http://www.slab.dnj.ynu.ac.jp/challenge2000.

[16] M. Tan. Cost-sensitive learning of classification knowledge and its application in robotics. *Machine Learning*, 13(1):7–33, 1993.

[17] K. M. Ting. A comparative study of cost-sensitive boosting algorithms. In *Proc. of the 17th International Conference on Machine Learning (ICML)*, 2000.

[18] T. N. Tran and M. Numao. Mining medical data with the assistance of biomedical literature sources. Technical Report SIG-KBS-A201, Japanese Society for Artificial Intelligence, 2002. In Japanese.

[19] S. Tsumoto. The common medical data sets to compare and evaluate KDD methods. *Journal of Japanese Society for Artificial Intelligence*, 15(5):751–758, 2000.

[20] P. D. Turney. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, 2:369–409, 1995.

[21] H. Yokoi, K. Takabayashi, Y. Satomura, S. Hirano, and S. Tsumoto. Introduction to the hepatitis dataset. In *Proc. of ECML/PKDD2002 Workshop on Discovery Challenge*, 2002.