Web for Data Mining: Organizing and Interpreting the Discovered Rules Using the Web

Yiming Ma
School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543
maym@comp.nus.edu.sg

Bing Liu
School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543
liub@comp.nus.edu.sg

Ching Kian Wong
School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543
wongck@comp.nus.edu.sg

ABSTRACT

The web not only contains a vast amount of useful information, but also provides a powerful infrastructure for communication and information sharing. In this paper, we present a system (called DS-Web) that uses the web to help data mining. Specifically, we use the web to facilitate delivering and interpreting the discovered rules. Interpreting the discovered rules to gain a good understanding of the domain is an important phase of data mining. It is also a very difficult task because the number of rules involved is often very large. This problem has been regarded as a major obstacle to the use of data mining results. DS-WEB assists the user in understanding a set of discovered rules in two steps. First, it finds a special subset (or a summary) of the rules that represents the essential relationships of the domain to build a hierarchical structure of the rules. It then publishes this hierarchy of rules via multiple web pages connected using hyperlinks. By using the web, we inherit the advantages of the web, e.g., accessibility, multi-user communication and friendly interface. DS-WEB not only allows the user to browse the rules easily, but also allows us to create a virtual workspace where multiple users can share opinions on the rules. This ultimately contributes towards comprehension of the domain. Our application experiences show that DS-WEB is much more powerful than a conventional system.

Keywords

Rule understanding, the Web, post-processing, interestingness, association rules.

1. INTRODUCTION

Interpreting the discovered rules to gain a good understanding of the domain is one of the important phases of the KDD process. It usually requires the user to browse a large set of discovered rules. Typical techniques (commonly called post-processing techniques) that assist the user in the process include templates [9], expectations [11, 21], summarization [13], and visualization [20].

In this paper, we focus on using the web to help the user to interpret a set of association rules. Finding interesting/useful knowledge from a set of association rules is a particularly hard problem as the number of rules is usually very large. This is because association rule mining aims to discover all rules that exist in the data. Furthermore, there is no logical ordering of the rules other than a ranking based on the confidence or the support level of the rules. Using confidence and support might not be a good way of organizing the set of rules, as they are not directly related to how interesting a rule is to the user. It is, however, very important to interpret and to understand the large set of rules since it contains the complete knowledge of the domain [1, 2].

One way of overcoming the problem is to employ some form of summarization so that only the essential subset of the rules are presented to the user at the beginning. Several research efforts have been focused on trying to find a summarized subset of the rules. In [13], the authors proposed a novel method to summarize the complete set of rules to only the essential relationships in the domain. The rules in this subset are called *direction-setting rules* (DS rules). In this paper, we propose to use DS rules to organize the discovered rules into a hierarchical structure to facilitate browsing. At the top of the hierarchy are the DS rules with bottom levels made up of following non-DS rules. Non-DS rules provide additional details of the domain. This technique will be discussed in section 3.

In recent years, the Internet has become the single most important medium for communication and information sharing. Among all its applications, the world-wide-web is the most popular. The web owes its popularity to its accessibility and also its intuitive interface. We can easily inherit some of the web's advantages if we can deliver our rules for post-analysis via the web.

1) Contents published on the web can be accessed from any computer in the world. This eliminates the need for a distant party to re-execute the mining system in order to get access to the same rules.
2) The web offers a familiar interface environment to browse the discovered rules. This vastly reduces any learning curve that might otherwise be required for other formats like graphs or maps of a conventional interface.

Rules can be easily published as web pages since rules, in its natural form, are already sentence-like. Our proposed hierarchical structure can be implemented by utilizing hypertext links to simulate the next level of rules. Users can start from the top level DS rules and then interactively drill down to access the non-DS rules. This aids the analysis and interpretation process.

Besides offering a web interface for easy browsing of the rules, the Internet also provides a basic infrastructure for co-operative work to be performed through its interconnected network. Users can individually browse, evaluate and comprehend the rules while browsing the web pages and at the same time choose to share whatever comments, insights or ideas about the rules with others. This creates a virtual multi-user workspace. It is both logical and realistic to have multiple users to analyze the complete set of rules because

(1) many users may be interested in the same set of rules (each

may be interested in a different subset), and

(2) the size of the rule set could overwhelm a single user, and

(3) by combining the expertise of different users, we could eliminate any bias a single user might have towards certain domain knowledge.

In this paper, we present our web-based post-processing framework and the DS-WEB system that implements the framework. The paper is organized as follows: Section 2 discusses the related work. Section 3 describes the direction setting rules used in building our hierarchical structure. Section 4 presents the proposed framework and the DS-WEB system. Section 5 gives an account of our application experiences. Section 6 concludes the paper.

2. RELATED WORK

Several commercial products offer the capability of performing post-processing of discovered association rules. Association rule miners in, e.g. SGI MineSet [20], IBM Intelligent Miner [8], Clementine [5], DBMiner [6] etc., present discovered association rules either in a graphical or text form. Typically, the association rules are ranked by their confidence and support levels. This usually led the user to focus only on the high confidence and high support rules. However, high confidence or support level may not imply high importance or interestingness of a rule. We feel that this simple strategy does not fully explore the potentials of the discovered association rules. Users should be allowed to browse and to analyze the rules more comprehensively without being overwhelmed by too many rules. Our framework proposes to use direction-setting rules to summarize the discovered rules and then use them to create a logical hierarchical structure so that users can interactively drill down to an interesting aspect.

There are a number of reported research works on the topic of finding interesting association rules in the KDD literature [e.g., 21, 11, 12, 13, 14, 17]. The focus is to use some interestingness measures to prune the discovered associations and/or to use the user's domain knowledge to help him/her to identify interesting rules. We can easily adapt these algorithms into our proposed framework. Since this paper mainly focuses on how to use the web to deliver the discovered rules and to help the user identify interesting rules, we will not discuss these works further. Interested readers, please refer to [e.g., 21, 11, 12, 13, 14, 17]. DS-WEB basically uses the summarization technique in [13] to organize the discovered rules. It then employs the web to believer this organization for user browsing and understanding.

Many existing systems have adopted a visual approach to analyzing discovered knowledge, most notably, the MineSet system from SGI [20]. In MineSet, an innovative method is used to offer a visual tour of a wide variety of concept descriptions (e.g. decision trees, decision tables, regression trees, scatter graphs, etc). Users can interact with these visual models to incrementally extract interesting knowledge. Our work promotes a similar idea of allowing users to interact directly with the concept description for incremental understanding. However, we also actively help the user by doing post-processing, i.e. summarization and organization. In addition, our framework builds an infrastructure to allow for collaborative work between users via the web.

There are also a number of vendors that offer web-based analysis tools. JWAVE [22] from Visual Numerics is a web-based software that can be used to produce visually comprehensive models of a company's data. It differs from our work in that it analyses attributes but not concept descriptions discovered from a mining process. WebSemba [3] from ASOC also utilizes the web as an infrastructure for delivering derived knowledge. However, it focuses on utilizing the knowledge and not on performing post-processing in order for the user to understand the domain.

3. SUMMARIZING THE DISCOVERED ASSOCIATIONS

In this section, we present a systematic way of summarizing the discovered associations. The technique summarizes the associations to a small set of *direction setting rules* (DS rules). The DS rules give the essential behavior of the discovered associations. In other word, they represent the essential relationships or structure of the domain. The non-DS rules simply give additional details. Using the DS rules as a summary, the user can interactively focus on the key aspects of the domain and selectively view the relevant details (non-DS rules).

3.1 General ideas

In this work, we focus on association rule mining from a relational table, which consists of a set of records described by a number of attributes. An item is an attribute value pair, i.e., (attribute = value) (numeric attributes are discretized). Association rule mining in such data is typically targeted at a specific attribute because the user normally wants to know how other attributes are related to this *target* attribute (which can have many values) [12, 4]. With a target attribute, we can express an itemset as follows (instead of a set of items as in [1, 2]):

$$X \rightarrow y$$

where y is an item (or a value) of the target attribute, and X is a set of items from the rest of the attributes. For simplicity, we call this itemset a *rule* hereafter, regardless of whether it is significant or not. We also say a rule is *large* if it meets the minimum support.

The proposed technique first prunes the discovered associations to remove those insignificant ones, and then finds a special subset of the unpruned associations to form a summary of the discovered associations. In the following subsection, we use an example to briefly introduce how we summarize the set of unpruned associations. The detailed descriptions of the technique and also how pruning is performed can be found in [13].

3.2 Summarizing the unpruned rules

Pruning can reduce the number of rules substantially. However, the number of rules left can still be very large. We then find a subset of the rules, called direction-setting rules (or DS rules), to summarize the unpruned rules. Essentially, DS rules are significant association rules that set the directions for non-DS rules to follow. The direction of a rule is the type of correlation it has, i.e., *positive correlation* or *negative correlation* or *independence*, which is computed using χ^2 test (see [13]). Let us see an example.

Example: We have the following discovered rules:

R1: Job = yes → Loan = approved
[sup = 40%, conf = 70%]

R2: Own_house = yes → Loan = approved
[sup = 30%, conf = 75%]

χ^2 analysis shows that having a job is positively correlated to the grant of a loan, and owning a house is also positively correlated to obtaining a loan. Then, the following association is not so surprising to us:

R3: Job = yes and Own_house = yes → Loan = approved
[sup = 20%, conf = 90%]

because it intuitively follows R1 and R2. We can use R1 and R2 to provide a summary of the three rules. R1 and R2 are DS rules as they set the direction (positive correlation) that is followed by R3. In real-life data sets, a large number of associations are like R3.

From the example, we see that the DS rules give the essential relationships of the domain. The non-DS rule is not surprising if we already know the DS rules. However, this, by no means, says that non-DS rules are not interesting. Non-DS rules can provide further details about the domain. For example, the non-DS rule above (R3) gives a higher confidence, which may be of interest to the user. Using DS rules to form a summary is analogous to summarization of a text article. From the summary, we know the essence of the article. If we are interested in the details of a particular aspect, the summary can point us to them in the article. In the same way, the DS rules give the essence of the domain and points the user to those related non-DS rules. Non-DS rules are combinations of DS rules. In the above example, R3 is a combination of R1 and R2.

Experiment results, including real-life applications, show that although the number of discovered rules can be huge, the number of DS rules is very small [13]. They can be analyzed manually by the human user to obtain the essential relationships in the data. He/she can then focus his/her attention on those interesting aspects of the relationships, and to see the relevant non-DS rules. The user can now obtain a complete picture of the domain without being overwhelmed by a huge number of rules.

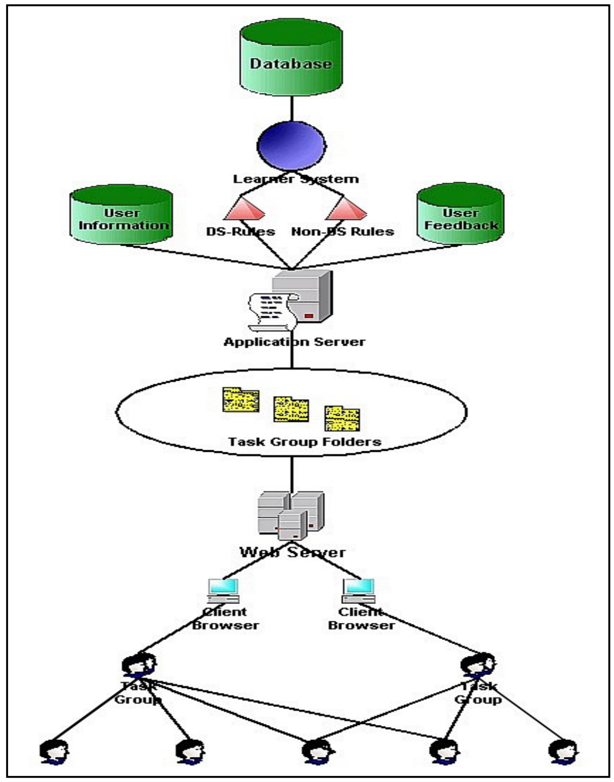
4. DS-WEB

In this section, we present the web part of DS-WEB. The framework exploits the web-based client/server architecture. It basically consists of a set of dynamically generated web pages that describe the discovered knowledge. These pages can be mounted, through a web server, onto either the Internet or a corporate intranet to enable sharing and collaboration between users. In the rest of this section, we introduce our server program in the sub-section 4.1, and client program in the sub-section 4.2.

4.1 Server program

Our server-client architecture is given in Figure 1. There are two servers in the diagram:

1) Application Server: Responsible for organizing the mining results and the user information.
2) WEB server: Responsible for presenting the mining results to different groups of users.



(Figure 1. High-level view of DS-WEB)

4.1.1 Application server

Application server mainly organizes the following three types of information:

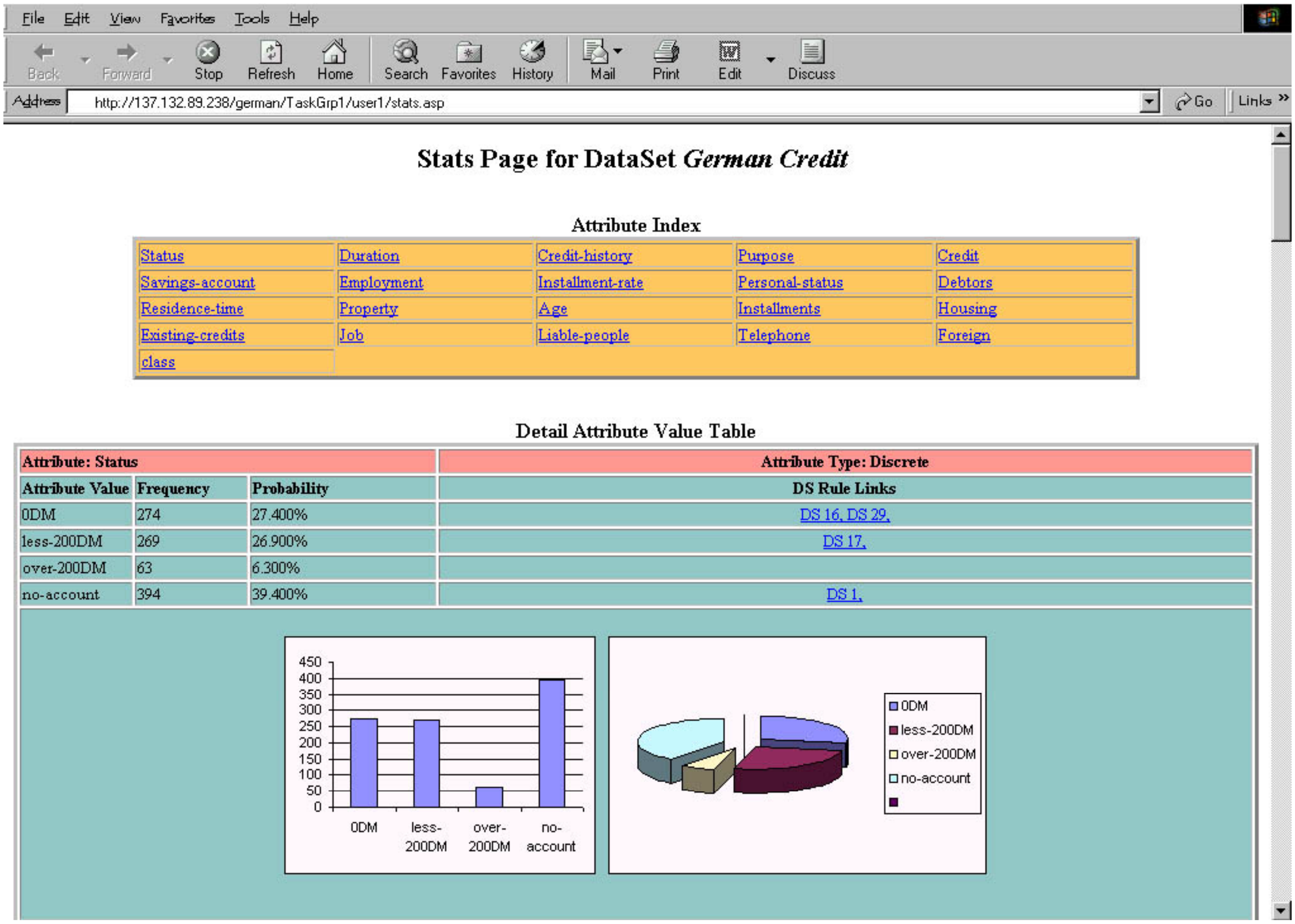
1) User information: Different users have different access rights to the mining results. On the server, we define the task groups, and store them in a user information database.
2) DS rules and non-DS rules (mining results): Based on the task groups, the server further filters the rules, and set up *task group folders* (a virtual folder to dynamically provide the relevant DS, non-DS rules to different task groups).
3) User feedback information: A user in a task group can write his/her comments on the rules and also rate the rules. These are kept in a database, and organized by the server.

4.1.2 Web server

The group folder basically stores a set of rule indexes to the rule databases. Based on the requests from different task groups, Web server dynamically retrieves the relevant rules from the respective task group folder and presents them as a set of HTML pages to the users.

4.1.3 Implementation details

We implemented our server control programs on the Windows NT server platform. SQL server controls the rule, feedback and user databases. SQL server makes the concurrent rule access possible for the multiple users environment. The Internet Information Server (IIS) publishes the Active Server Page (ASP) to the



(Figure 2. Attribute distribution page)

respective users. DB-WEB uses the following database tables:

1) Application table: It keeps all the ongoing applications. It has three attributes:
   - Application ID: It is the primary key.
   - Application name
   - Application start time

2) Rule set table: It keeps all the rules (DS, and non-DS). It has five attributes:
   - Rule ID: it is the primary key for other tables to refer
   - Application ID: it keeps track of different applications.
   - Bit flag: it is used to differentiate the DS and non-DS rules. We index this attribute to achieve fast DS and non-DS rules access.
   - Support and confidence: it records down the support and confidence measures.

3) DS and non-DS dependency table: It keeps relations among DS and non-DS rules. It has two attributes:
   - DS rule ID: foreign key to the Rule set table
   - Non-DS rule ID: foreign key to the Rule set table. It is a follower of the DS rule

4) Attribute table: It keeps all the attribute information, and user access privileges. It has three attributes:
   - Attribute ID: it is the primary key.
   - Application ID: it is a foreign key of the application table. It shows this attribute belongs to which application.
   - Attribute name
   - User access string: it links to the User Group and User tables to check who has access to this attribute.

5) Attribute value table: It keeps all the attribute value information. It has four attributes
   - Attribute value ID: it is the primary key.
   - Attribute value name
   - Attribute ID: it is the foreign key to the Attribute table.
   - Frequency: it records the number of appearances of each attribute value.

6) User Group Information table: It keeps all the group information. It has three attributes:
   - Group ID: it is the primary key.
   - Group name
   - Application access string: it is used to check the applications that this group is working on.

7) User Information table: It keeps all the user information. It has two attributes:
   - User ID: it is the primary key.
   - User Group String: it marks which group the user

belongs to, and also the position in the group.

8) Comments table: It keeps all the comments from users. It has three attributes:
   - Comments ID: it is a primary key.
   - User ID
   - Rule ID: it is a foreign key to the Rule set table. We index on this field to speed up query processing.
   - Update Time
   - Ratings
   - Comments String

With this set of database tables, we can dynamically manage the mining results in a multi-user environment. In the rest of this section, we discuss DS-WEB from a user's (client) point of view.

4.2. Client program

Once the client program connects to the server, all the accessible task group folders are shown to the users based on the Application table (see section 4.1.3). Users can open these task group folders, and retrieve the relevant rules from each folder. Based on the DS-rules and non-DS rules on the WEB server, we present a set of web pages to users. In the following sub-sections, we introduce the process for a systematic analysis and understanding of the association rule space.

4.2.1 Browsing the attribute distribution

Attribute distribution information can be very useful to the users. From this information, the user is able to get a basic feel of the data. All our real-life applications showed that users would like to understand the attribute distribution before other mining results are presented.

Therefore, the first web page presented to the user is the attribute distribution information of the database. We dynamically pull out the relevant attribute information from the Attribute and Attribute value tables (see section 4.1.3). Simple graph plots are provided to help users understand the attribute distributions. The user also can export the information to other statistical packages.

More importantly, this page provides a starting point to the DS-rules. After analyzing the attribute distributions, the user can directly go to the DS rules, and start to explore the relevant knowledge. All the DS-rules are indexed. The set of DS-rule links are properly placed under relevant attribute-value pairs or conditions (Figure 2, previous page) (the sample used in this section is taken from *German Credit* dataset [15]).

4.2.2 Exploring a DS-rule and its non-DS rules

To help users understand a rule (DS or non-DS), we present our rules in two forms, one for expert users and the other for novice users. The expert rule format (Figure 3) presents a rule in a conjunctive form. The advantage of this format is that expert users can view the relevant information quickly. In Figure 3, the first number within the brackets is the support, the second is the confidence, the third is the support count of the conditions and the fourth is the support count of the conditions and the consequent. However, a novice user might encounter problems in understanding the rule in this form. To overcome this, we offer a novice format, which is presented in a more understandable form (Figure 4). By making use of the support and confidence information from the rule, we rewrite the rule, and add in some qualitative measures (i.e. Large portion of data, High confidence etc., which are suggested by domain experts).

As we mentioned earlier, non-DS rules are the followers of the DS rules with more conditions and higher confidences. Some of the high confidence non-DS rules could also be important for understanding the domain. We use the HTML page (Figure 5, next page) to explore the non-DS rules. This information can be obtained by joining DS rules ID with DS and Non-DS dependency table.

Expert format		Novice format
Housing = own Employment = over-seven-years → class = Loan Approved (17.200%, 100.000%, 172, 172)	⟹	A **large portion** of data (17.200%) having value: Housing = own Employment = over-seven-years leads to: class = Loan Approved with **very high confidence 100.000%** compare to the distribution of the: class = Loan Approved of 70.000%
(Figure 3. Rule format for expert)		(Figure 4. Rule format for novice)

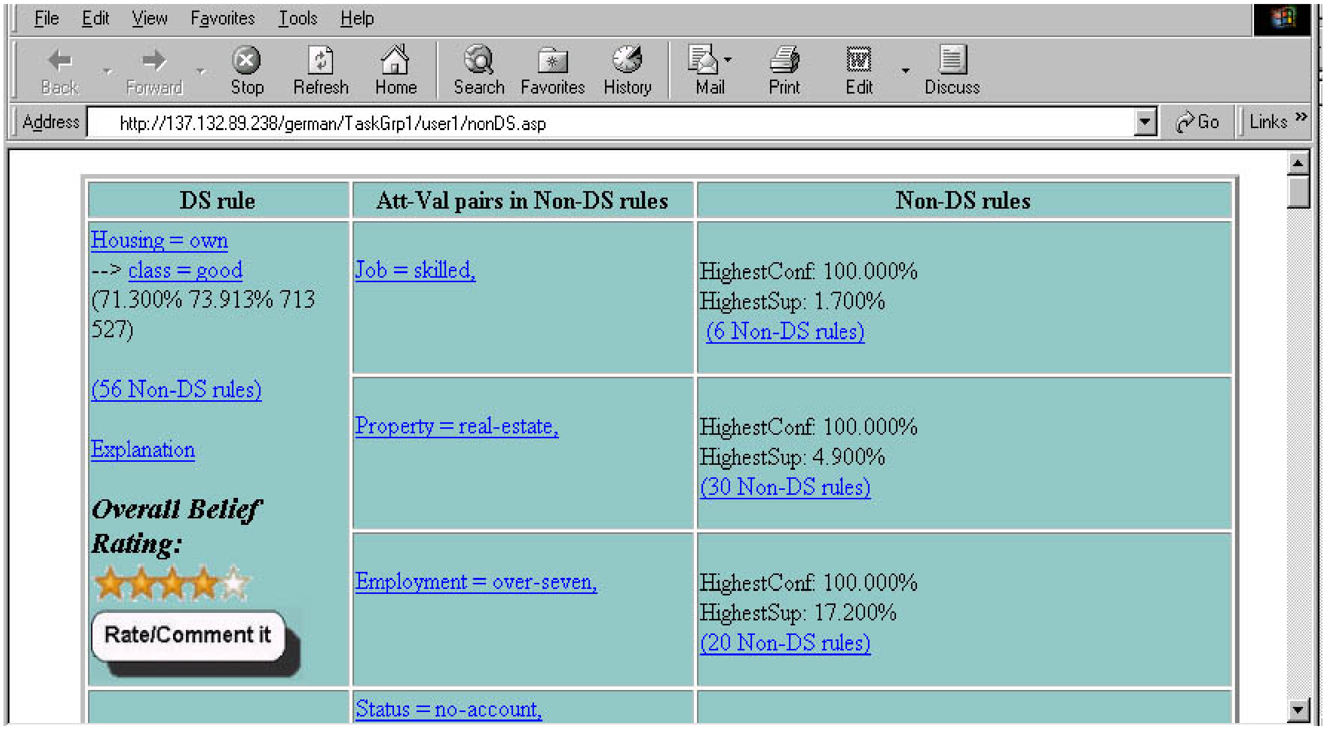
In column 1 of the sample HTML page of Figure 5, we present the DS rules in the expert format (Figure 3). The user can directly jump to the relevant non-DS rules from the non-DS rule hyperlinks. He/she can also switch to the novice rule format (Figure 4) by following the *Explanation* hyperlink. At the bottom of each DS rule, we provide the overall user *belief rating* of the rule. This rating gives the user some ideas of how believable this rule is to other users in the same task group. A user can see the detailed ratings and give his/her own views of the rule by clicking the *Rate/Comment it* button, which leads to Figure 6 (see the next sub-section).

From our application experiences, the users typically would like to focus on a few important attributes in exploring the non-DS rules. Hence, in column 2 of Figure 5, we present the additional conditions (attribute value pairs) that non-DS rules have. They give a rough knowledge of what conditions are involved in the non-DS rules. The user can easily choose the interesting conditions to start off.

Column 3 (Figure 5) contains a hyperlink to the set of all such non-DS rules. We also let the user know how many such non-DS rules there are in total and give the highest confidence, and support level in of the non-DS rules. By viewing the highest confidence and support information, users will have some ideas on the quality of the non-DS rules involved.

It is also interesting to understand the relationship between DS rules. In our system, we group the similar DS rules together. By carefully comparing and contrasting the DS-rules, we could provide the user more useful information.

On top of these features, we also have rule search utilities to help the user search for the relevant rules that match certain criteria.



(Figure 5. Navigating non-DS rules)

4.2.3 Collaboration features

In DS-WEB, we construct a virtual web workspace where users can share opinion on the perceived believability and the interestingness/usefulness of a particular rule. That is, users can rate each DS-rule or non-DS rule, and log their comments. We can retrieve the rating of a rule by joining Rule ID with Comments table (see section 4.1.3). Such features are not only useful in a multi-user environment but also useful in a single user case. Figure 6 (in the next page) gives a screen shot of the web-based review (rating and commenting) form.

In DS-WEB, we allow two types of rating to be given by users.

1). *Belief rating*: The user uses this rating to express his/her opinion on the believability of the rule.

2). *Interestingness rating*: The user uses this rating to indicate his/her interest in the rule. This information is used in collaborative filtering (see below).

In Figure 6, column 1 lists each individual rule and its overall consolidated belief rating together with information about the number of users that have contributed to the rating. This gives the current user an idea of the perceived quality of the rule at one glance.

Details of how each individual user rated this rule are listed in column 2. Users are also allowed to attach annotations or comments to each rule. This serves as an explanation of why a particular rating is given to a rule. Links to these comments also appear in column 2. Column 3 provides a means for the user to submit his/her own belief rating, interestingness rating and comments on a rule. Submitted information is consolidated by the application server and becomes visible to the other users. Note that the interestingness rating of a user is only used by the system (see below) and not revealed to others users. The system also provides an option to apply different weights to ratings from different users, based on certain discriminating criteria like seniority or user expertise in a particular area.

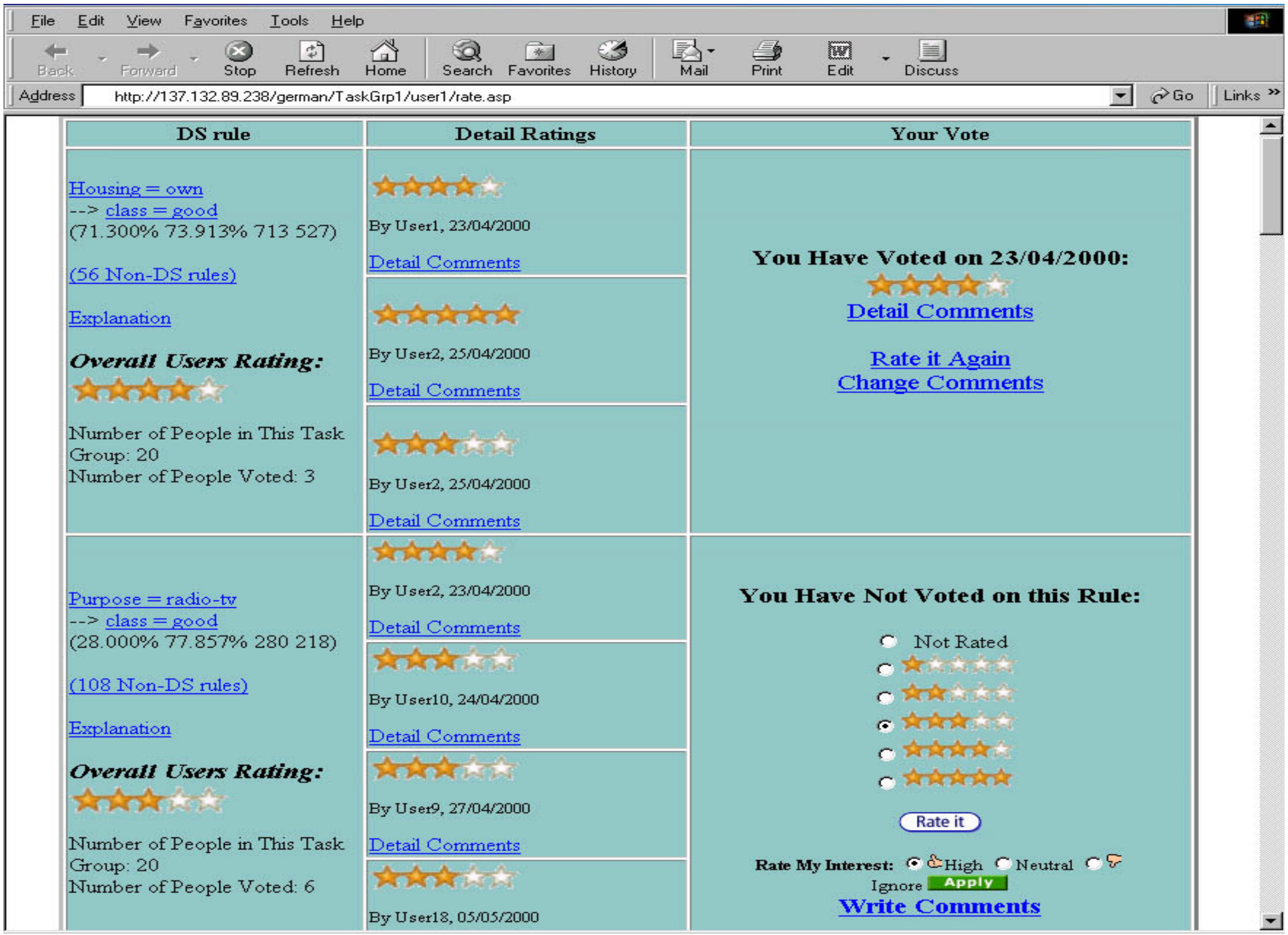
The user can also request for a listing of rules sorted based on either his/her own or on overall consolidated ratings (belief or interestingness ratings). A sorted list of rules based on the user's own ratings reminds the user of his/her previous works while a sorted list of rules based on overall consolidated ratings gives the user an idea of how others in the task group think. The system can compare a user's rating with the overall rating given by others in the group on the different rules to give a *Conformity Level*. This shows the user how inline his/her views are as compared to the group.

The interestingness rating (Figure 7, also in Figure 6) not only allows the user to record down those interesting rules, but also allows the system to perform collaborative filtering, i.e., the system can suggest a list of other rules that might be of interest to the user based on the records of others with the same interest.

(Figure 7. Indicating interest level to a rule)

5. APPLICATIONS OF DS-WEB

We have used our system in a number of real-life applications. We are still enhancing the system to suit the users' needs. In this section, we briefly describe one of these applications we have



(Figure 6. WEB based review form)

done for an education institution.

This institution needs to analyze its students across the years. One of the tasks is to study the profile of the students. We have the complete set of student data. There are 45 attributes. It includes education, family background, sex, ages, etc. In this application, it is not sufficient to just analyze the statistical distribution of each attribute. The institution is more interested in correlation among the attributes. More importantly, they would like to have a good understanding of their students.

When we run the normal association rule-mining engine, it generates a large number of associations (more than 20,000), which are very difficult for the users to understand. When we run our DS-WEB on this set of data, the number of DS rules generated is 138, and non-DS rules is 421. It now becomes possible for the human users to interpret them manually.

There are 4 users working on this project. Since they are all familiar with the WEB interface, they can use our system without much difficulty. They only used less than an hour to select the relevant rules, and started to write their comments on the rules. By using our review form (Figure 6), they easily gathered the opinions from one another. After working for half a day, they identified 17 DS rules and 45 non-DS rules that are important for understanding the domain. The collaboration features of our system make it easy for users to communicate with each other.

In this application, we find that the WEB interface is very useful. Users are familiar with it, and can work in their offices using a web browser without installing any software systems on their computer. Based on our experiences, we are now working on a more generic and powerful system.

6. CONCLUSION

This paper proposes a framework to aid the user in analyzing a set of association rules. The set of rules is first organized into a hierarchical structure using DS rules to facilitate browsing. The framework then uses the web as an infrastructure to create a virtual workspace where multi-users can co-operatively work towards evaluating rules in the rule set to filter out noisy and imprecise knowledge by sharing opinions. This contributes significantly towards the final comprehension of the discovered knowledge. Although this paper focus primarily on association rules, this framework can also be applied to other concept descriptions with little modification.

We envision that not only will the web be the primary channel for knowledge delivery in the near future but much of the KDD process will also shift towards a distributed multi-participant environment. Performing post-processing as proposed in our framework becomes very natural.

ACKNOWLEDGEMENTS

This project is funded by National Science and Technology Board, and National University of Singapore under RP3981678.

REFERENCES

[1]. Agrawal, R., Imielinski, T., Swami, A. "Mining association rules between sets of items in large databases." *Proceedings of ACM SIGMOD International Conference on Management of Data* (*SIGMOD-1993*), 1993.

[2]. Agrawal, R. and Srikant, R. 1994. "Fast algorithms for mining association rules." *Proceedings of International Conference on Very Large Data Bases* (*VLDB-94*), 1994.

[3]. http://www.asoc.com/ehome.html

[4]. Bayardo, R., Agrawal, R, and Gunopulos, D. "Constraint-based rule mining in large, dense databases." *Proceedings of IEEE International Conference on Data Engineering* (*ICDE-99*), 1999.

[5]. http://www.spss.com/clementine/

[6]. http://www.dbminer.com/

[7]. Fayyad, U. M., Irani, K. B. "Multi-interval discretization of continuous-valued attributes for classification learning." *Proceedings of the International Joint Conference on Artificial Intelligence* (*IJCAI-93*), 1022-1027.

[8]. http://www-4.ibm.com/software/data/iminer/

[9]. Klemetinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A.I. "Finding interesting rules from large sets of discovered association rules." *Proceeding of the International Conference on Information and Knowledge Management* (*CIKM-1994*), 1994.

[10]. Kohavi, R., John, G., Long, R., Manley, D., and Pfleger, K. 1994. MLC++: a machine learning library in C++. *Tools with artificial intelligence*, 740-743.

[11]. Liu, B. and Hsu, W. 1996. Post-analysis of learned rules. *of the Thirteenth National Conference on Artificial Intelligence* (*AAAI-96*), *AAAI-96*, 1996.

[12]. Liu B., Hsu W. and Ma Y. "Integrating Classification and Association Rule Mining." *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining* (*KDD-98*), 1998

[13]. Liu B., Hsu W. and Ma Y. "Pruning and Summarizing the Discovered Associations." *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD-99*), 1999.

[14]. Liu, B., Hu, M. and Hsu, W. "Multi-level organization and summarization of the discovered rules," To appear in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (*KDD-2000*), 2000.

[15]. Merz, C. J, and Murphy, P. UCI repository of machine learning databases. [http://www.cs.uci.edu/~mlearn/MLRepository.html].

[16]. Park, J. S., Chen, M-S, and Yu, P. S. "An Effective Hash Based Algorithm for Mining Association Rules" *Proceedings of ACM SIGMOD International Conference on Management of Data* (*SIGMOD-95*), 1995.

[17]. Padmanabhan, B., and Tuzhilin, A. "A belief-driven method for discovering unexpected patterns." *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining* (*KDD-98*), 1998.

[18]. Quinlan, J. *C4.5: programs for machine learning*. Morgan Kaufmann, 1992.

[19]. Srikant, R. and Agrawal, R. "Mining generalized association rules." *Proceedings of the International Conference on Very Large Data Bases* (*VLDB-1995*), 1995.

[20]. http://www.sgi.com/software/mineset/

[21]. Silberschatz, A., and Tuzhilin, A. "On Subjective Measure of Interestingness in Knowledge Discovery" *Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (*KDD-95*), 1995

[22]. http://www.vni.com/products/wpd/jwave/index.html

About the authors

Yiming Ma (http://www.comp.nus.edu.sg/~maym) is a MSc student and a senior tutor at School of Computing, National University of Singapore. He received his B.Sc. (first class honours) from National University of Singapore. His honours project was selected as the best project, for which he received "IEEE (Singapore) Information Technology Book Prize 1998-1999." In pursuing his MSc degree in Computer Science, he received "Dean's Graduate Award" for his excellent research work. He expects to receive his MSc degree later this year.

Bing Liu (http://www.comp.nus.edu.sg/~liub) is an Associate Professor at School of Computing, National University of Singapore. His research interests include knowledge discovery and data mining, text and web mining, and constraint satisfaction. He received his B.Sc. degree from Northwest Institute of Light Industry, China, and his Ph.D. degree from University of Edinburgh, U.K. Before he joined National University of Singapore, he worked as a member of technical staff at National Computer Board, Singapore. Dr. Liu has published many research papers in prestigious conferences and journals (e.g., SIGKDD, AAAI, IJCAI, and IEEE transactions) in the areas of data mining, constraint satisfaction, and Artificial Intelligence. He is a member of American Association for Artificial Intelligence (AAAI), IEEE Computer Society, and ACM SIGKDD.

Ching Kian Wong is a MSc student and a research assistant at School of Computing, National University of Singapore. He received his B.Sc. from National University of Singapore. He expects to complete his MSc thesis later this year.