View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Queensland University of Technology ePrints Archive

Mining for Useful Association Rules Using the ATMS

Yue Xu and Yuefeng Li
School of Software Engineering and Data Communications
Queensland University of Technology
Brisbane, Queensland 4001, Australia
{yue.xu,y2.li}@qut.edu.au

Abstract

Association rule mining has made many achievements in the area of knowledge discovery in databases. Recent years, the quality of the extracted association rules has drawn more and more attention from researchers in data mining community. One big concern is with the size of the extracted rule set. Very often tens of thousands of association rules are extracted among which many are redundant thus useless. In this paper, we first analyze the redundancy problem in association rules and then propose a novel ATMS-based method for extracting non-redundant association rules.

1. Introduction

Association rule mining has become one of the most important and well researched techniques of data mining. It aims to extract interesting correlations and associations among sets of items in large datasets. Two phases are involved in mining association rules: extracting frequent itemsets and generating association rules from the frequent itemsets with the constraints of minimal confidence. The rules whose confidence is larger than a user-specified minimum confidence threshold are considered interesting or useful. Various approaches have been proposed, most of them are focused on developing novel algorithms and data structures to aid efficient computation of such rules [1, 6], especially on improving the efficiency of generating frequent itemsets. However, the quality of the mined association rules hasn't drawn adequate attention. As a matter of fact, very often the resulting rule base can easily contain several thousands of rules among which are many redundancies and thus useless in practice. While some efforts have been done on reducing the size of the extracted rule base by defining various interest measures, incorporating constraints into mining process, or designing specific templates to mine for restricted rules [4, 5, 8], the useless redundancy in the extracted rule base is still a problem to solve.

The first phase of association rule mining generates all frequent itemsets. The second phase can be viewed as a process of adding new extracted association rules into the current rule set in an accumulating manner. The extracted rules work as the beliefs that represent the epistemic state of the decision making system of the domain. From the logic point of view, when adding a new extracted rule into the rule set, the reasoning system needs to maintain a consistent and non-redundant rule set. The techniques that enable a reasoning system to update its belief set in dealing with new information is called belief revision. Many approaches have been proposed in the area of belief revision. Among them, the truth maintenance systems (ATMS) provide a powerful mechanism.

In this paper, a preliminary investigation of applying the technique of the Assumption-based Truth Maintenance System (ATMS) [3] to association rule mining is presented. Next section will discuss some related approaches. The task of association rule mining and the redundancy problem in association rules are discussed in Section 3. Section 4 gives a brief review to the ATMS. In Section 5, we present the algorithms that generate non-redundant association rules based on the ATMS technique. Section 6 concludes the paper.

2. Related Work

One approach to address the quality of association rules is to apply constraints to generate only those association rules that are interesting to users based on some constraints instead of all the association rules. [5] and [8] proposed some algorithms that incorporate item constraints to the process of generating frequent itemsets. Item constraints restrict the items and combination of items that are interesting according to users, association rules are generated from those frequent itemsets. Also some works have been done on measuring association rules with interestingness parameters. These approaches focus on pruning the association rules to get more general or informative association

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE
IEEE COMPUTER SOCIETY

rules based on interestingness parameters. The approach proposed in [2] integrates various constraints into mining process including consequent constraint and minimal improvement constraint as well. The consequent constraint is used to restrict rules with certain consequent specified by the user, while minimal improvement constraint is used to simplify the antecedents of rules based on item's contribution to the confidence and therefore prune association rules that have more specific antecedent but do not make more contribution to the confidence. Another approach is to use a taxonomy of items to extract generalized association rules [4], i.e., to generate rules between itemsets that belong to different abstract levels in the taxonomy, especially between high abstract levels aiming at reducing the number of extracted rules.

The approaches mentioned above aim at reducing the number of extracted rules, but eliminating redundancy of rules is not a focus. The approaches proposed in [9] and [7] focus on extracting non-redundant rules. Both of them make use of the closure of the Galois connection to extract non-redundant rules between frequent closed itemsets insdead of rules between frequent itemsets. Their results show that any rule between itemsets is equivalent to some rule between closed itemsets. Thus many redundant rules can be eliminated. One difference between the two approaches is the definition of redundant rules. [9] prefers both shorter antecedent and shorter consequent among rules which have the same confidence, while [7] defines non-redundant rules are those which have minimal antecedent and maximal consequent. Our definition of non-redundant rules is similar to that of [7]. But we use a very different mechanism to prune the redundant rules. Most importantly, our method does not have to calculate the closed itemsets, while the method proposed in [7] must calculate closed itemsets and the generators of the closed itemsets as well.

3. Association Rule Mining Problems

3.1. Association Rules

Let $I = \{I_1, I_2, \ldots, I_m\}$ be a set of m distinct items, T be transaction that contains a set of items such that $T \subseteq I$, D be a database containing different identifiable transactions. An association rule is an implication in the form of $X \Rightarrow Y$, where $X, Y \subset I$ are sets of items called itemsets. The rule means that X implies Y. Various metrics describe the utility of an association rule. The most common ones are the percentage of all transactions containing $X \cup Y$ which is called the support, and the percentage of transactions containing Y among transactions containing X which is called confidence of the rule. Association rule mining is to find out association rules that satidfy the predefined minimum support and confidence from a given database.

TID	Items
1	A C D
2	B C E
3	A B C E
4	B E
5	A B C E
6	B C E

Table 1. A simple database

1-itemsets	2-itemsets	3-itemsets	4-itemsets
$\{A\}$ 3/6	$\{AB\}$ 2/6	$\{ABC\}$ 2/6	$\{ABCE\}$ 2/6
$\{B\}$ 5/6	$\{AC\}$ 3/6	$\{ABE\}$ 2/6	
$\{C\}$ 5/6	$\{AE\}$ 2/6	$\{ACE\}$ 2/6	
$\{E\}$ 5/6	$\{BC\}$ 4/6	$\{BCE\}$ 4/6	
	$\{BE\}$ 5/6		
	$\{CE\}$ 4/6		

Table 2. Frequent itemsets (minisupp=2/6)

The problem is usually decomposed into two subproblems. One is to find those itemsets whose occurrences exceed a predefined threshold in the database, those itemsets are called frequent itemsets. This step is computationally intensive. The second subproblem is to generate association rules from those frequent itemsets with the constraints of minimal confidence. This step is relatively straightforward. Rules of the form $X \Rightarrow Y \setminus X$ are generated for itemset Y, for all $X \subset Y$ and $X \neq \phi$. We should consider each subset X of Y as an antecedent except for the empty and the full itemset. The example in [7] will be used to explain our approach throughout the paper. The simple database is given in Table 1 which contains 6 transactions. The frequent itemsets generated from the database (minisupp=2/6) are shown in Table 2. 50 association rules which are given in Table 3 are extracted from these frequent itemsets if the minimum confidence threshold is set to 2/6.

3.2. Redundancy in Association Rules

The rules in Table 3 are considered useful based on the predefined minimum confidence. However, after a careful inspection of the rules, we can find that some of the rules actually do not contribute new information to the rule set, i.e., without these rules the rule set doesn't lose any information. For example, rules 1, 13, 14, and 15 can be derived from rule 37 and all these rules have the same confidence. Therefore, removing rules 1, 13, 14, and 15 won't change the context of the rule set. All these rules have the same antecedent and identical confidence, while rule 37 has the longest consequent which is a superset of the consequent of other rules and thus brings more information than the other rules. Therefore, we consider rules 1, 13, 14, and 15 are redundant to rule 37. On the other hand, rules 25, 32, and 47 have the same consequent as rule 6 does but have a longer

0-7695-2504-0/05 $20.00 © 2005 IEEE
IEEE COMPUTER SOCIETY

Rule No.	Rules	Rule No.	Rules
1	$A \Rightarrow B$ 2/3	26	$AB \Rightarrow C$ 1
2	$A \Rightarrow C$ 1	27	$AC \Rightarrow B$ 2/3
3	$A \Rightarrow E$ 2/3	28	$AC \Rightarrow E$ 2/3
4	$B \Rightarrow A$ 2/5	29	$AE \Rightarrow C$ 1
5	$B \Rightarrow C$ 4/5	30	$AE \Rightarrow B$ 1
6	$B \Rightarrow E$ 1	31	$BC \Rightarrow A$ 1/2
7	$C \Rightarrow A$ 3/5	32	$BC \Rightarrow E$ 1
8	$C \Rightarrow B$ 4/5	33	$BE \Rightarrow A$ 2/5
9	$C \Rightarrow E$ 4/5	34	$BE \Rightarrow C$ 4/5
10	$E \Rightarrow A$ 2/5	35	$CE \Rightarrow B$ 1
11	$E \Rightarrow B$ 1	36	$CE \Rightarrow A$ 1/2
12	$E \Rightarrow C$ 4/5	37	$A \Rightarrow BCE$ 2/3
13	$A \Rightarrow BE$ 2/3	38	$B \Rightarrow ACE$ 2/5
14	$A \Rightarrow CE$ 2/3	39	$C \Rightarrow ABE$ 2/5
15	$A \Rightarrow BC$ 2/3	40	$E \Rightarrow ABC$ 2/5
16	$B \Rightarrow AE$ 2/5	41	$AB \Rightarrow CE$ 1
17	$B \Rightarrow CE$ 4/5	42	$AC \Rightarrow BE$ 2/3
18	$B \Rightarrow AC$ 2/5	43	$AE \Rightarrow BC$ 1
19	$C \Rightarrow AB$ 2/5	44	$BC \Rightarrow AE$ 1/2
20	$C \Rightarrow BE$ 4/5	45	$BE \Rightarrow AC$ 2/5
21	$C \Rightarrow AE$ 2/5	46	$CE \Rightarrow AB$ 1/2
22	$E \Rightarrow AC$ 2/5	47	$ABC \Rightarrow E$ 1
23	$E \Rightarrow BC$ 4/5	48	$ABE \Rightarrow C$ 1
24	$E \Rightarrow AB$ 2/5	49	$ACE \Rightarrow B$ 1
25	$AB \Rightarrow E$ 1	50	$BCE \Rightarrow A$ 1/2

Table 3. Association rules (miniconf=2/6)

antecedent to be satisfied than rule 6 needs. That means, rules 25, 32, and 47 do not bring more information but require more in order to be fired. In this case, we consider rules 25, 32, and 47 are redundant to rule 6. The following definition defines such kind of redundant rules.

Definition 3.1 (Redundant rules) *Let $X \Rightarrow Y$ and $X' \Rightarrow Y'$ are two association rules which have the same confidence. $X \Rightarrow Y$ is said a redundant rule to $X' \Rightarrow Y'$ if $X' \subseteq X$ and $Y \subseteq Y'$.*

The following propositions can help to prune out most redundant rules.

Proposition 3.1 *Let $r_1 : X \Rightarrow Y_1$ and $r_2 : X \Rightarrow Y_2$ be two association rules, r_1 and r_2 have the same antecedent. If conf(r_1) = conf(r_2) and $Y_1 \subset Y_2$, then, $\forall \alpha \in 2^{\beta} \setminus \beta$, $\alpha \Rightarrow \beta \setminus \alpha$ is redundant, where $\beta = (Y_1 \cap Y_2) \cup X$ and 2^{β} is the power set of β.*

The following example illustrates the correctness of the proposition. Suppose we have generated rules 15 and 37 in Table 3, i.e., $A \Rightarrow BC$ and $A \Rightarrow BCE$. Since the two rules have the same confidence, we get the following equation:

$$\frac{supp(ABC)}{supp(A)} = \frac{supp(ABCE)}{supp(A)}$$

It's easy to find that $\forall \alpha \in \{A, B, C, AB, AC, BC\}$ which is the power set of $\{A, B, C\} \cap \{A, B, C, E\}$ without the intersection set, the following equation is true:

$$\frac{supp(ABC)}{supp(\alpha)} = \frac{supp(ABCE)}{supp(\alpha)}$$

According to the definition of the redundant rules, $\alpha \Rightarrow ABC \setminus \alpha$ is redundant to $\alpha \Rightarrow ABCE \setminus \alpha$ because the two rules have the same antecedent and confidence, and the consequent of the first rule is shorter than that of the second rule. This gives the result that $A \Rightarrow BC$, $B \Rightarrow AC$, $C \Rightarrow AB$, $BC \Rightarrow A$, $AC \Rightarrow B$, and $AB \Rightarrow C$ are redundant to the rules $A \Rightarrow BCE$, $B \Rightarrow ACE$, $C \Rightarrow ABE$, $BC \Rightarrow AE$, $AC \Rightarrow BE$, and $AB \Rightarrow CE$, respectively.

Following Definition 3.1, we have the proposition 3.2. The correctness of this proposition can be proved directly by Definition 3.1.

Proposition 3.2 *Let $r_1 : X_1 \Rightarrow Y$ and $r_2 : X_2 \Rightarrow Y$ be two association rules, r_1 and r_2 have the same consequent. If conf(r_1) = conf(r_2), $X_1 \subset X_2$, then the rule $X_2 \Rightarrow Y$ is redundant to the rule $X_1 \Rightarrow Y$.*

These propositions will not be helpful if we don't have an efficient way to identify rules with identical antecedents or identical consequents. In the following sections, we present a method that solves this problem by using truth maintenance techniques.

4. A Review to de Kleer's ATMS

As mentioned above, the ATMS [3] provides an attrictive mechanism to maintain and update the belief set. The ATMS itself doesn't take part in any problem solving. Whenever the problem solving system draws a conclusion or wants to add a new rule, it passes the conclusion or the rule to the ATMS. The ATMS then incorporates the new information into the belief set and maintains consistent and minimal (non-redundant) environments (to be introduced below) for each datum supplied by the problem solver. Based on the belief set and the dependency among data nodes, the problem solver makes further inferences towards a solution to the domain problem. Even though an association rule mining system is not a typical problem solving system that uses an ATMS to confirm its inference conclustions, the ATMS techniques can be used in association rule mining for avoiding redundancy in the extracted rules. Section 5 will present an approach that combines the propositions discussed in Section 3 with the basic ATMS algorithm to extract non-redundant rules. In this section we will briefly descrebe the ATMS [3].

- Node: A node in an ATMS represents any datum suplied by the problem solver. The In the association rule mining system presented in this paper, each datum is an itemset.

- Assumptions: The problem solver specifies a set of distinguished nodes which are called assumptions. The assumptions are presumed to be true by the problem solver. In the case of association rule mining, all the items form the set of assumptions.

- Justifications: A justification is an implication of the form $a_1 \wedge a_2 \ldots \wedge a_n \Rightarrow b$ where each a_i is a node. Justifications are supplied by the problem solver. The intent of the justification is to inform the ATMS that when the antecedent nodes are derivable, then the consequent node is also derivable. In the association rule mining algorithm present in this paper, each justification is a potential association rule generated by the algorithm.

- Environments: A set of assumptions is called an environment of a node if the node holds under this environment. An environment in the association rule mining system is an itemset.

- Label: The label of a node contains all environments of the node. Each environment in the label is a non-redundant set of assumptions from which the node can be derived.

- Context: The context of an environment contains all nodes which can be derived from the environment.

- Nogood: There is a special node called Nogood in an ATMS. Any environment in which contradiction is derived is included in the label of Nogood.

The ATMS is provided with a set of assumptions and justifications. The task of the ATMS is to efficiently determine the label for each node and context for each environment. In an ATMS, each node is associated with a label and a set of justifications denoted as $< node, label, justifications >$. Both the label and justifications for a node can be explained as material implications. Given a node n with label $\{\{A_1, A_2, \ldots,\}, \{B_1, B_2, \ldots,\}, \ldots\}$ and justifications $\{(x_1, x_2, \ldots), (y_1, y_2, \ldots) \ldots\}$, the meaning of the label of n is that the conjunction of assumptions in each environment makes n true. In general, each justification is non-redundant. That is, deleting any element in a justification will destroy the implication relation of this justification to its node. For any two justifications of a node, usually these two justifications don't imply each other. If one justification can be inferred from the other, then the effect of this justification will be covered by the latter one. The same rules also apply to the environments for a node. That is, any environment is non-redundant and any two environments of a node have at least one different assumption. These are the important features of the ATMS that we will use to find non-redundant association rules.

de Kleer didn't give a formal semantics of the ATMS. The behaviour of the ATMS is described by operational procedures. Usually, supplied with a set of justifications Σ and a proposition p, the ATMS is required by the problem solver to determine whether or not p is believed in Σ. The ATMS is incremental. After processing all the justifications in Σ, the $label(p)$ in node γ_p provides all the environments under which p can be believed. If $label(p) = \{\}$, then p cannot be believed.

The main ATMS algorithm **label-update**$(a_1, \ldots, a_k \Rightarrow p)$ is to update the labels of nodes with the addition of a justification. Before the algorithm is called, it is supposed that the labels of each antecedent node as well as the consequent node of the justification have already been created. Σ is the set of justifications currently in the ATMS. The algorithm **label-update** is described as follows.

Algorithm: label-update$(a_1, \ldots, a_k \Rightarrow p)$

1. $L_p := \{x \mid x := \bigcup_{i=1}^{i=k} x_i, \quad x_i \in label(a_i)\}$
2. **for** each environment $e \in L_p$ **do**
3. **if** e contains any Nogood
   then remove e from L_p.
4. **if** e subsumes any environments in $label(p)$
   then remove e from L_p.
5. **endfor**
6. **if** p is $\perp$ **then** $label(p) := L_p$;
   $nogood := nogood \bigcup L_p$; **return**.
7. **if** $L_p \neq label(p)$ **then** $label(p) := label(p) \cup L_p$
   add $a_1, \ldots, a_k \Rightarrow p$ as a justification of p
8. **for** each justification $r \in \Sigma$ such that p is one antecedent of r, **label-update**(r).
9. **endif**

Algorithm **label-update(p)** is a recursive algorithm. In order to maintain a consistent data base, after updating the label of p, the algorithm will propagate the new label of p to other propositions which take p as one of their antecedents.

5. Association Rule Mining based on the ATMS

Let Γ be the set of all frequent itemsets. The second subproblem in association rule mining can be viewed as a search problem that searches the space $\mathcal{S} \subseteq \Gamma \times \Gamma$, where $\mathcal{S} = \{< X, Y > | X, Y \in \Gamma, X \subset Y\}$, for a subspace $\gamma \times \gamma \subseteq \mathcal{S}$ so that each $< X, Y > \in \gamma \times \gamma$ is considered a useful association, i.e., $X \Rightarrow Y \setminus X$ is a useful rule, and each $< X', Y' > \in \Gamma \times \Gamma \setminus \gamma \times \gamma$ is considered useless. The traditional rule generation method examines each point $< X, Y >$ in $\mathcal{S}$ to determine the associations that satisfy the minimum confidence constraint. The method is straightforward, but the resulting association rules contain a great amount of redundancy. This section presents a method that examines the most promising points in a dynamicly shrinking space for non-redundant association rules. The method uses a simple heuristic to guide the search and applies the ATMS technique to help prune redundancy. According to Definition 3.1, the rules with shorter antecedent and longer consequent have better

0-7695-2504-0/05 $20.00 © 2005 IEEE
IEEE COMPUTER SOCIETY

chance to be non-redundant. Based on this heuristic, the point $< X, Y >$ in $\mathcal{S}$ which has the highest value of $Y.size/X.size$ is examined first, and the size of $\mathcal{S}$ keeps reduced as more redundant rules are identified and removed from $\mathcal{S}$. $X.size$ denotes the number of items in X.

As discussed in Section 4, the label of a node in the ATMS provides all antecedents of the node and the context of an environment provides all consequents of the environment. If the confidence of any two elements in the label of a node or in the context of an environment is recognized identical, one of the two propositions can be used to prune out redundant rules. One of the limitations of the original ATMS is that it can only infer results with absolute truth values. It cannot represent a plausible conclusion with a degree of belief. In most cases, labels of nodes are very complicated and probability distributions on assumptions are not independent. This makes the probability calculation very difficult. Fortunately, in the case of association rule mining, we can use the supports of itemsets to calculate the confidence of each rule straightway. The algorithms are described below which take $\mathcal{S}$ as input and produce a set of non-redundant association rules as output. Let $h(X, Y)$ denote $Y.size/X.size$.

Algorithm: ATMS_ARM

Input: $\mathcal{S}$

Output: set of justifications Σ

1. $\Sigma := \phi$
2. **while** $\mathcal{S} \neq \phi$
3. r:= $X \Rightarrow Y \setminus X, h(X,Y) = \max_{<a,b> \in \mathcal{S}} \{h(a,b)\}$
4. $\Sigma := \Sigma \cup \{r\}$
5. **labelupdate-ARM**(r)
6. Remove r from $\mathcal{S}$
7. **endwhile**

Algorithm **labelupdate-ARM** modified the **labelupdate** of the original ATMS by incorporating confidence calculation and redundancy pruning using the two propositions. In the algorithm below, $e.supp$ and $power(e)$ denote the support of itemset e and power set of e, respectively. It is assumed that the supports have been calculated before executing the algorithm. Each element e in a label $label(p)$ is represented as a pair: $e : conf$, meaning that $conf$ is the confidence of rule $e \Rightarrow p$.

Algorithm: labelupdate-ARM$(a_1, \ldots, a_k \Rightarrow p)$

1. **if** node(p) or node(a_i) doesn't exist, $i = 1, \ldots, k$
2. **then** create node(p) with $lable(p) = \{\{p : 1\}\}$
   create node(a_i) with $lable(a_i) = \{\{a_i : 1\}\}$
3. $L_p := \{x : c \mid x := \bigcup_{i=1}^{i=k} x_i,$
   $c := (p \cup x).supp/x.supp,$
   $x_i : c_i \in label(a_i)\}$
4. **for** each environment $e : c \in L_p$ **do**
5. **if** $c < minconf$ **then** remove $e : c$ from L_p
6. **endfor**
7. **if** $L_p = lable(p)$ or $L_p = \phi$ **then**
8. $\Sigma := \Sigma \setminus \{a_1, \ldots, a_k \Rightarrow p\}$;
   $\mathcal{S}$:= $\mathcal{S} \setminus < a_1 \wedge \ldots \wedge a_k, a_1 \wedge \ldots a_k \wedge p >$;
   return
9. $label(p) := label(p) \cup L_p$
10. **if** $\exists e_i{:}c_i \in label(p)$ and $e_i \cap p \neq \phi$ **then**
11. remove $e_i : c_i$ from $label(p)$
12. **if** $\exists e_i{:}c_i, e_j{:}c_j \in label(p)$, c_i=c_j, $e_i \subset e_j$
13. **then**
    $\mathcal{S}$:=$\mathcal{S} \setminus < e_j, p \cup e_j >$
    $\Sigma := \Sigma \setminus \{e_j \Rightarrow p\}$
14. **for** each $e : c \in label(p)$ **do**
15. $context(e) := context(e) \cup \{p : c\}$
    if $\exists p_i, p_j \in context(e)$, $c_i = c_j$, and $p_i \subset p_j$,
    then
    for each $\alpha \in power(\beta) \setminus \beta$,
    $\beta := (p_i \cap p_j) \cup e$ **do**
    $\mathcal{S}$:=$\mathcal{S} \setminus < \alpha, \beta >$
    $\Sigma := \Sigma \setminus \{\alpha \Rightarrow \beta \setminus \alpha\}$
16. **endfor**
17. **endfor**
18. for each justification $r \in \Sigma$ such that p is one antecedent of r, **labelupdate-ARM**(r).

In the algorithm above, step 7 is to check whether the newly added rule makes any contribution to the existing label. If it doesn't, the newly added rule should be removed since what it concludes can be implied by other rules. Step 10 is to remove implications of the form $A \Rightarrow AB$ that are not considered valid accoring to the definition of association rules but might be deduced by the propagation of labels. Steps 12 to 13 and steps 14 to 17 implement the propositions 3.2 and 3.1, respectively.

The algorithm proposed here focuses only on redundancy pruning but doesn't address the problem of consistency check. For classical logical reasoning, a contradiction is a conflict among absolute truth values. But, in the case of plausible reasoning like association rule mining, the truth value of a statement is with a degree of uncertainty rather than absolute true or false. The consistency check for plausible reasoning is much more complicated than in the case of classical logical reasoning. This issue is important, but little effort has been made. This topic will be disscused in a forthcoming paper.

The effectiveness and efficency of the algorithm can be demonstrated by the example dataset mentioned in Section 3. Initially, the search space $\mathcal{S}$ contains 50 elements. Based on the heuristic value h, these elements are examined in the following order:

0-7695-2504-0/05 $20.00 © 2005 IEEE
IEEE COMPUTER SOCIETY

$< A, ABCE >, < B, ABCE >, < C, ABCE >, < E, ABCE >, < A, ABC >, < A, ABE >, < A, ACE >, \ldots, < E, BCE >, < A, AB >, \ldots, < E, CE >, < AB, ABCE >, \ldots, < CE, ABCE >, < AB, ABC >, \ldots, < CE, BEC >, < ABC, ABCE >, \ldots, < BCE, ABCE >$. After examining the first 5 points, we get the following results:
$label(A) = \{A : 1\}$, $label(BCE) = \{A : 2/3\}$, $label(B) = \{B : 1\}$, $label(ACE) = \{B : 2/5\}$, $label(C) = \{C : 1\}$, $label(ABE) = \{C : 2/5\}$, $label(E) = \{E : 1\}$, $label(ABC) = \{E : 2/5\}$, $label(BC) = \{A : 2/3\}$, $context(B) = \{ACE : 2/5\}$, $context(C) = \{ABE : 2/5\}$, $context(E) = \{ABC : 2/5\}$, $context(A) = \{BCE : 2/3, BC : 2/3\}$
$context(A)$ indicates that two consequents of A have the same confidence and one is a subset of the other. Acording to Proposition 3.1 (step 17 in Algorithm labelupdate-ARM), six elements are removed from $\mathcal{S}$: $< A, ABC >, < B, ABC >, < C, ABC >, < AB, ABC >$, $< BC, ABC >$, and $< AC, ABC >$. Similarly, examining $< A, ABE >$ and $< A, ACE >$ makes another 12 points removed from $\mathcal{S}$ which include: $< A, ABE >, < B, ABE >, < E, ABE >, < AB, ABE >, < AE, ABE >, < BE, ABE >, < A, ACE >, < C, ACE >, < E, ACE >, < AC, ACE >, < AE, ACE >$, and $< CE, ACE >$. That is, after examining the first 7 points in $\mathcal{S}$, 18 rules are eliminated, and the search space $\mathcal{S}$ is therefore reduced by 36%. This reduction is significant not only because the reduced space makes the search more efficient but also because the elimination of redundant rules greatly improves the quality of the extracted association rules.

The result is shown in Table 4. Out of the 50 potential rules, 33 redundant rules are eliminated. 17 rules are extracted and considered usefull. But two of them, rules 45 and 50 which are indicated with $*$ in Table 4, are actually redundant but missed to be caught. The main reason is because the algorithm takes the advantage of the ATMS that helps examine only the rules with the same antecedent or consequent.

6. Conclusion

The quality of extracted association rules is getting more and more attention in the area of association rule mining. One problem with this concern is the presence of redundancy in the extracted association rules. This paper presented a novel method that applies the truth maitenance technique ATMS to the association rule mining. The basic idea of the method is to take the advantage of the ATMS's labels and contexts to easily identify the redundancy among association rules. Especially, this method allows us to find non-redundant rules without having to calculate the closed itemsets. Two features of the method allow potential extensions to the current model. One is the possibility to incorporate constraints to antecedents or consequents by means of the labels and contexts. This can restrict the rules to the ones whose antecedents or consequents satify some constraints specified by the users. The other feature is the use of NOGOOD to check the consistency among the extracted rules.

Rule No.	Extracted Rules	Redundant Rules
2	$A \Rightarrow C$ 1	26, 29, 48
6	$B \Rightarrow E$ 1	32,47
11	$E \Rightarrow B$ 1	25, 30, 35,49
41	$AB \Rightarrow CE$ 1	
43	$AE \Rightarrow BC$ 1	
17	$B \Rightarrow CE$ 4/5	
20	$C \Rightarrow BE$ 4/5	8, 9
23	$E \Rightarrow BC$ 4/5	12
37	$A \Rightarrow BCE$ 2/3	1, 3,13,14,15,27,28,42
7	$C \Rightarrow A$ 3/5	
44	$BC \Rightarrow AE$ 1/2	31
46	$CE \Rightarrow AB$ 1/2	36
50^*	$BCE \Rightarrow A$ 1/2	
38	$B \Rightarrow ACE$ 2/5	4,5,16, 18,33,34
39	$C \Rightarrow ABE$ 2/5	19,21
40	$E \Rightarrow ABC$ 2/5	10,22,24
45^*	$BE \Rightarrow AC$ 2/5	

Table 4. Extracted Rules and Redundant Rules

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on very large data bases*, pages 487–499, 1994.

[2] R. J. Bayardo, R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. *Data Mining and Knowledge Discovery*, 4:217–240, 2000.

[3] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[4] J. Han and Y. Fu. Mining multiple-level association rules in large databases. *IEEE Transactions on Knowledge and Data Engineering*, 11:798–804, 5 2000.

[5] R. T. NG, V. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning otimizations of constrained association rules. In *Proceedings of the SIGMOD conference*, pages 13–24, 1998.

[6] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th ICDT conference*, pages 398–416, 1999.

[7] N. Pasquier, R. Taouil, Y. Bastide, G. Stumme, and L. Lakhal. Generating a condensed representation for association rules. *Journal of Intelligent Information Systems*, 24(1):29–60, 2005.

[8] R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. In *Proceedings of the KDD Conference*, pages 67–73, 1997.

[9] M. J. Zaki. Generating non-redundent association rules. In *Proceedings of the KDD Conference*, pages 34–43, 2000.

0-7695-2504-0/05 $20.00 © 2005 IEEE

IEEE COMPUTER SOCIETY