Journal of Intelligent Information Systems, 22:2, 155–174, 2004

Mining Informative Rule Set for Prediction

JIUYONG LI
Department of Mathematics and Computing, University of Southern Queensland, Toowoomba, QLD 4350, Australia

HONG SHEN* shen@jaist.ac.jp
Graduate School of Information Science, Japan Advanced Institute of Science and Technology, Tatsunokuchi, Ishikawa 923-1292, Japan

RODNEY TOPOR
School of Computing and Information Technology, Griffith University, Nathan, Qld 4111, Australia

Received December 27, 2001; Revised November 26, 2002; Accepted April 11, 2003

Abstract. Mining transaction databases for association rules usually generates a large number of rules, most of which are unnecessary when used for subsequent prediction. In this paper we define a rule set for a given transaction database that is much smaller than the association rule set but makes the same predictions as the association rule set by the confidence priority. We call this rule set informative rule set. The informative rule set is not constrained to particular target items; and it is smaller than the non-redundant association rule set. We characterise relationships between the informative rule set and non-redundant association rule set. We present an algorithm to directly generate the informative rule set without generating all frequent itemsets first that accesses the database less frequently than other direct methods. We show experimentally that the informative rule set is much smaller and can be generated more efficiently than both the association rule set and non-redundant association rule set.

Keywords: association rule, data mining, prediction

1. Introduction

1.1. Introduction

The rapidly growing volume and complexity of modern databases makes the need for technologies to describe and summarise the information they contain increasingly important. The general term for this process is data mining. Association rule mining is the process of generating associations or, more specifically, association rules, in transaction databases. Association rule mining is an important area of data mining and has wide application in many fields. Two key problems in association rule mining are the high cost for generating association rules and the large number of rules generated. Much work has been done to address the first problem. Methods for reducing the number of rules

*Author to whom all correspondence should be addressed.

generated depend on applications, because a rule may be useful in one application but not another.

In this paper, we are particularly concerned with generating rules for prediction. For example, given a set of association rules that describe the shopping behavior of the customers in a store over time, and some purchases made by a particula customer, we wish to predict what other purchases will be made by that customer.

The association rule set (Agrawal et al., 1993) can be used for prediction if the high cost for finding and applying the rule set is not a concern. The constrained and optimality association sets (Bayardo and Agrawal, 1999; Bayardo et al., 1999) can not be used for this prediction because their rules do not have all possible items to be consequences. The non-redundant association rule set (Zaki, 2000) may be used, but can be large in size. Our task of this work is to find a more effective way for prediction.

The general method for generating association rules by first generating frequent itemsets can be unnecessarily expensive, as many frequent itemsets do not result in useful association rules. For the purpose of effective prediction, we define the informative (association) rule set that is smaller than the association rule set and makes the same predictions by the confidence priority, and present a direct method for generating the informative rule set that does not involve generating frequent itemsets first. Unlike other algorithms that generate rules directly, our method imposes no constraints on the consequences of generated rules as did in Bayardo and Agrawal (1999) and Bayardo et al. (1999) and accesses the database less frequently than other unconstrained methods (Webb, 2000).

1.2. Related work

Association rule mining was first studied in Agrawal et al. (1993). Most research work has been done on how to mine frequent itemsets efficiently. Apriori (Agrawal and Srikant, 1994) is a widely accepted approach, and there have been many enhancements to it (Holsheimer et al., 1995; Houtsma and Swami, 1995; Mannila et al., 1994; Park et al., 1995; Savasere et al., 1995). In addition, other approaches have been proposed (Han et al., 2000; Shenoy et al., 1999; Zaki et al., 1997), mainly by using more memory to save time. For example, the algorithm presented in Han et al. (2000) organizes a database into a condensed structure to avoid repeated database accesses, and the algorithms in Shenoy et al. (1999) and Zaki et al. (1997) use the vertical layout of databases to save counting time.

Some direct algorithms for generating association rules without generating frequent itemsets first have also been proposed (Bayardo et al., 1999; Bayardo and Agrawal, 1999; Webb, 2000). Algorithms presented in Bayardo et al. (1999) and Bayardo and Agrawal (1999) focused only on one fixed consequence and hence is inefficient for mining all association rules. The algorithm presented in Webb (2000) needs to scan a database as many times as the number of all possible antecedents of rules. As the result, it may not be efficient when a database is too large to be retained in the memory.

There are two types of algorithms to simplify the association rule set, namely direct and indirect algorithms. Most indirect algorithms simplify the set by post-pruning and reorganization, as in Toivonen et al. (1995), Liu et al. (1999) and Ng et al. (1998), which can obtain an association rule set as simple as the user wishes but does not improve the efficiency of rule mining process. There have been some attempts to simplify the association rule set directly. The algorithm for mining constraint rule sets is one such attempt (Bayardo et al., 1999). It produces a small rule set and improves the mining efficiency by pruning unwanted rules in the process of rule mining. However, a constraint rule set contains only rules with some specific items as consequences, as do the optimality rule sets (Bayardo and Agrawal, 1999). They are not suitable for association prediction where all items may be consequences. The most significant work in this direction is to mine the non-redundant rule set because it simplifies the association rule set and retains the information intact (Zaki, 2000). However, the non-redundant rule set is still too large to be used effectively for prediction.

1.3. Our contributions

The main contributions of this paper are listed as follows:

We define a new rule set, namely informative rule set, for a given transaction database, which is the smallest rule set presenting the same prediction as the association rule set by confidence priority. We characterise its relationships with the non-redundant association rule set.

We present a direct algorithm to generate the informative rule set efficiently. The algorithm generates rules at the same time when generating frequent itemsets. Unlike other direct association rule mining algorithms, the proposed algorithm accesses the database less frequently for generating rules on all possible items.

We compare the informative rule set with constrained and optimality association rule sets, and characterise the relationships between the informative association rule set and non-redundant association rule set.

We show experimentally on standard synthetic data that the informative rule set is much smaller and can be generated more efficiently than both association rule set and non-redundant rule set.

2. The informative rule set

2.1. Association rules and related definitions

Let $I = \{1, 2, \ldots, m\}$ be a set of *items*, and $T \subseteq I$ be a *transaction* containing a set of items. An *itemset* is defined to be a set of items, and a k-itemset is an itemset containing k items. A database D is a collection of transactions. The *support* of an itemset (e.g. X), denoted by $sup(X)$, is the ratio of the number of transactions containing the itemset to the number of all transactions in a database. Given two itemsets X and Y where $X \cap Y = \emptyset$, an association rule is defined to be $X \Rightarrow Y$, where $sup(X \cup Y)$ and $sup(X \cup Y)/sup(X)$ are not less than user specified thresholds respectively and $sup(X \cup Y)/sup(X)$ is called the *confidence* of the rule, denoted by $conf(X \Rightarrow Y)$. The two thresholds are called the *minimum support* and the *minimum confidence* respectively. For convenience, we abbreviate $X \cup Y$ by XY and use the terms rule and association rule interchangeably in the rest of this paper.

Suppose that every transaction is given a unique identifier. A set of identifiers is called a *tidset*. Let mapping $t(X)$ be the set of identifiers of transactions containing the itemset X. It is clear that $sup(X) = |t(X)|/|D|$. In the following, we list some basic relationships between itemsets and tidsets.

1. $X \subseteq Y \Rightarrow t(X) \supseteq t(Y)$,
2. $t(X) \subseteq t(Y) \Rightarrow t(XZ) \subseteq t(YZ)$ for any Z, and
3. $t(XY) = t(X) \cap t(Y)$.

We say that rule $X \Rightarrow Y$ is *more general* than rule $X' \Rightarrow Y$ if $X \subset X'$, and we denoted this by $X \Rightarrow Y \subset X' \Rightarrow Y$. Conversely, $X' \Rightarrow Y$ is *more specific* than $X \Rightarrow Y$. We define the *covered set* of a rule to be the tidset of its antecedent. We say that rule $X \Rightarrow Y$ *identifies* transaction T if $XY \subset T$. We use Xz to represent $X \cup \{z\}$ and $sup(X \neg Z)$ for $sup(X) - sup(XZ)$.

2.2. *The informative rule set*

Let us consider how a user uses the set of association rules to make predictions. Given an input itemset and the association rule set, initiate the prediction set to be an emptyset. Select a matching rule with the highest confidence from the rule set, and then put the consequence of the rule into the prediction set. We say that a rule matches a transaction if its antecedent is a subset of the transaction. To avoid repeatedly predicting on the same item(s), remove those rules whose consequences are included in the prediction set. Repeatedly select the next highest confidence matching rule from the remaining rule set until the user is satisfied or there is no rule to select. The justification for choosing the confidence priority model will be presented in the discussion section.

We have noticed that some rules in the association rule set will never be selected in the above prediction procedure, so we will remove those rules from the association rule set and form a new rule set. This new rule set has exactly the same prediction power, same set of prediction items in the same order of generation as the association rule set. Here, we consider the order because the user may stop selection at any time, and we will guarantee to obtain the same prediction items in this case. In addition, the sequence reflects the priority among items in the prediction itemset.

Formally, given an association rule set R and an itemset P, we say that the *prediction* of P from R is a sequence of items Q. The sequence Q is generated by using the rules in R in descending order of confidence. For each rule r that matches P (i.e., for each rule whose antecedent is a subset of P), each consequence of r is added to Q. After having added a consequence to Q, all rules with this consequence are removed from R.

To exclude those rules that have never been used in the prediction, we present the following definition.

Definition 1. Let R_A be an association rule set and R_A^1 the set of single-target rules in R_A. A set R_I is *informative* over R_A if (1) $R_I \subset R_A^1$; (2) $\forall r \in R_I \nexists r' \in R_I$ such that $r' \subset r$ and $conf(r') \geq conf(r)$; and (3) $\forall r'' \in R_A^1 - R_I$, $\exists r \in R_I$ such that $r'' \supset r$ and $conf(r'') \leq conf(r)$.

The following result follows immediately.

Lemma 1. *There exists a unique informative rule set for any given rule set.*

Proof: Suppose that we have two informative rule sets R_1 and R_2 for the complete rule set R. If $R_1 \neq R_2$, we must have a rule r such that $r \in R_1 \wedge r \notin R_2$. Since r is excluded by R_2, there must be another rule $r' \in R_2$ such that $r' \subset r$ and $conf(r') \geq conf(r)$. Clearly, R_1 cannot be informative by the definition, regardless whether it includes r' or not, resulting in contradiction.

Consequently, there exists a unique informative rule set for a complete rule set. □

We give two examples to illustrate this definition.

Example 1. Consider the following small transaction database: $\{1 : \{a, b, c\}, 2 : \{a, b, c\}, 3 : \{a, b, c\}, 4 : \{a, b, d\}, 5 : \{a, c, d\}, 6 : \{b, c, d\}\}$. Suppose the minimum support is 0.5 and the minimum confidence is 0.5. There are 12 association rules (that exceed the support and confidence thresholds). They are $\{a \Rightarrow b(0.67, 0.8), a \Rightarrow c(0.67, 0.8), b \Rightarrow c(0.67, 0.8), b \Rightarrow a(0.67, 0.8), c \Rightarrow a(0.67, 0.8), c \Rightarrow b(0.67, 0.8), ab \Rightarrow c(0.50, 0.75), ac \Rightarrow b(0.50, 0.75), bc \Rightarrow a(0.50, 0.75), a \Rightarrow bc(0.50, 0.60), b \Rightarrow ac(0.50, 0.60), c \Rightarrow ab(0.50, 0.60)\}$, where the numbers in parentheses are the support and confidence respectively. Every transaction identified by the rule $ab \Rightarrow c$ is also identified by rule $a \Rightarrow c$ or $b \Rightarrow c$ with higher confidence. So $ab \Rightarrow c$ can be omitted from the informative rule set without losing predictive capability. This is achieved by using requirements (2) and (3) in Definition 1. Rule $a \Rightarrow b$ and $a \Rightarrow c$ provide predictions b and c with higher confidence than rule $a \Rightarrow bc$, so rule $a \Rightarrow bc$ can be omitted from the informative rule set. This is achieved by using requirement (1) in Definition 1. Other rules can be omitted similarly, leaving the informative rule set containing the 6 rules $\{a \Rightarrow b(0.67, 0.8), a \Rightarrow c(0.67, 0.8), b \Rightarrow c(0.67, 0.8), b \Rightarrow a(0.67, 0.8), c \Rightarrow a(0.67, 0.8), c \Rightarrow b(0.67, 0.8)\}$.

Example 2. Consider the rule set $\{a \Rightarrow b(0.25, 1.0), a \Rightarrow c(0.2, 0.7), ab \Rightarrow c(0.2, 0.7), b \Rightarrow d(0.3, 1.0), a \Rightarrow d(0.25, 1.0)\}$. Rule $ab \Rightarrow c$ may be omitted from the informative rule set as the more general rule $a \Rightarrow c$ has an equal confidence. Rule $a \Rightarrow d$, must be included in the informative rule set even though it can be derived by transitivity from rules $a \Rightarrow b$ and $b \Rightarrow d$. Otherwise, if it were omitted, item d could not be predicted from the itemset $\{a\}$, as the definition of prediction does not provide for reasoning by transitivity.

Now we present the main property of informative rule set.

Theorem 1. *Let R_A be an association rule set. Then the informative rule set R_I over R_A is the smallest subset of R_A such that, for any itemset P, the prediction sequence of P from R_I equals the prediction sequence of P from R_A.*

Proof: We will prove this theorem from two aspects. Firstly, a rule omitted by R_I does not affect prediction from R_A for any P. Secondly, a rule set omitting one rule from R_I cannot present the same prediction sequences as R_A for any P.

Firstly, we will prove that a rule omitted by R_I does not affect prediction from R_A for any P.

For a single-target rule r' omitted by R_I, there must be another rule r in R_I such that $r \subset r'$ and $conf(r) \geq conf(r')$. When r' matches P, so does r. If both rules have the same confidence, omitting r' does not affect prediction from R_A. If $conf(r) > conf(r')$, r' must be automatically omitted from R_A after r is selected and the consequence of r is included in the prediction sequence. So, omitting r' does not affect prediction from R_A.

For a multiple-target rule in R_A, e.g. $A \Rightarrow bc$, there must be two rules $A' \Rightarrow b$ and $A'' \Rightarrow c$ in R_I for $A' \subseteq A$ and $A'' \subseteq A$ such that $conf(A' \Rightarrow b) \geq conf(A \Rightarrow bc)$ and $conf(A'' \Rightarrow c) \geq conf(A \Rightarrow c)$. When rule $A \Rightarrow bc$ matches P, $A' \Rightarrow b$ and $A' \Rightarrow c$ do. It is clear that if $conf(A' \Rightarrow b) = conf(A' \Rightarrow c) = conf(A \Rightarrow bc)$, then omitting $A \Rightarrow bc$ does not affect prediction from R_A. If $conf(A' \Rightarrow b) > conf(A \Rightarrow bc)$ and $conf(A' \Rightarrow c) > conf(A \Rightarrow bc)$, rule $A \Rightarrow bc$ must be automatically omitted from R_A after $A' \Rightarrow b$ and $A'' \Rightarrow c$ are selected and item b and c are included in the prediction sequence. Similarly, we can prove that omitting $A \Rightarrow bc$ from R_A does not affect prediction when $conf(A' \Rightarrow b) > conf(A'' \Rightarrow c) = conf(A \Rightarrow bc)$ or $conf(A'' \Rightarrow c) > conf(A' \Rightarrow b) = conf(A \Rightarrow bc)$. So omitting $A \Rightarrow bc$ from R_A does affect prediction. Similarly, we can conclude that a multiple-target rule in R_A does not affect its prediction sequence.

Thus a rule omitted by R_I does not affect prediction from R_A.

Secondly, we shall prove the minimum property. Suppose that we omit one rule $X \Rightarrow c$ from R_I. Let $P = X$, there must be a position for c in the prediction sequence from R_A determined by $X \Rightarrow c$ because there is no other rule $X' \Rightarrow c$ such that $X' \subset X$ and $conf(X' \Rightarrow c) \geq conf(X \Rightarrow c)$. When $X \Rightarrow c$ is omitted from R_I, there may be two possible results for the prediction sequence from R_I. One is that item c does not occur in the sequence. The other is that item c is in the sequence but its position is determined by another rule $X' \Rightarrow c$ for $X' \subset X$ with smaller confidence than that for $X \Rightarrow c$. As the result, the two prediction sequences can not be the same.

Hence, the informative rule set is the smallest subset of R_A that provides the same predictions for any itemset P. □

Finally, we describe a property that characterises some rules to be omitted from the informative rule set.

We can divide the tidset of an itemset X into two parts on an itemset (consequence), $t(X) = t(XZ) \cup t(X \neg Z)$. The first part means a set of transactions containing both itemsets X and Z, and the second part means a set of transactions containing itemset X but not Z. If the second part is an empty set, then the rule $X \Rightarrow Z$ has 100% confidence. Usually, the smaller is $|t(X \neg Z)|$, the higher is the confidence of the rule. Hence, $|t(X \neg Z)|$ is very important in determining the confidence of a rule.

Lemma 2. *If $t(X \neg Z) \subseteq t(Y \neg Z)$, then rule $XY \Rightarrow Z$ does not belong to the informative rule set.*

Proof: Let us consider two rules, $XY \Rightarrow Z$ and $X \Rightarrow Z$.

We know that $conf(XY \Rightarrow Z) = s_1/(s_1 + r_1)$, where $s_1 = |t(XYZ)|$ and $r_1 = |t(XY \neg Z)|$, and $conf(X \Rightarrow Z) = s_2/(s_2 + r_2)$, where $s_2 = |t(XZ)|$ and $r_2 = |t(X \neg Z)|$.

$$r_1 = |t(XY\neg Z)| = |t(X\neg Z) \cap t(Y\neg Z)| = |t(X\neg Z)| = r_2.$$
$$s_1 = |t(XYZ)| \leq |t(XZ)| = s_2.$$

As a result, $conf(XY \Rightarrow Z) \leq conf(X \Rightarrow Z)$. Hence rule $XY \Rightarrow Z$ must be omitted by the informative rule set. □

This is an important property of the informative rule set, since it enables us to predict rules that cannot be included in the informative rule set in the early stage of association rule mining. We will discuss this in detail in Section 4.

3. Comparison with the non-redundant association rule set

It is clear that the informative rule set is different from the constraint (Bayardo et al., 1999) and optimality (Bayardo and Agrawal, 1999) rule sets, because they do not have all possible items to be consequences and subsequently cannot make the same predictions as the association rule set. The non-redundant rule set (Zaki, 2000) can make the same prediction as the association rule set, but it is larger than the informative rule set. We now discuss its relationship with the informative rule set.

To facilitate our discussion, we first restate non-redundant rules in a way that makes it easy to compare with our informative rule set.

Generally, we say that a rule is derivable if its confidence and support can be derived from other more general rules. More specifically, rule $X \Rightarrow Y$ is derivable if there is a set of rules $\mathcal{R}$ in which all rules are more general than rule $X \Rightarrow Y$, such that rule $X \Rightarrow Y$ and its support and confidence can be obtained from $\mathcal{R}$. For example, rule $ab \Rightarrow c(0.2, 0.7)$ can be derived from two rules $a \Rightarrow b(0.25, 1.0)$ and $a \Rightarrow c(0.2, 0.7)$. The numbers in brackets are the support and the confidence.

We give one type of derivable rules as follows.

Lemma 3. *If $t(X) \subseteq t(Y)$, then for any itemset Z rule $XY \Rightarrow Z$ and $Z \Rightarrow XY$ are derivable.*

Proof: Since $t(X) \subseteq t(Y)$, rule $X \Rightarrow Y$ is a 100% confidence rule and $sup(XZ) = sup(XYZ)$. As a result, $sup(XY \Rightarrow Z) = sup(X \Rightarrow Z)$ and $conf(XY \Rightarrow Z) = conf(X \Rightarrow Z)$. Consequently, rule $XY \Rightarrow Z$ can be derived from rules $X \Rightarrow Z$ and $X \Rightarrow Y$.

Similarly, rule $Z \Rightarrow XY$ can be derived from rules $Z \Rightarrow X$ and $X \Rightarrow Y$ and its confidence and support are the same as those of rule $Z \Rightarrow X$.

Consequently, $XY \Rightarrow Z$ and $Z \Rightarrow XY$ are derivable. □

The following lemma follows immediately:

Lemma 4. *Redundant rules given in Zaki* (2000) (*Theorem* 5 *and Theorem* 6) *are derivable rules.*

Proof: Detailed in Appendix. □

By comparison, the informative rule set excludes at least all derivable rules given in the above lemma.

Firstly, all derivable rules given in Lemma 3 are omitted by the informative rule set. Since the confidence of rule $XY \Rightarrow Z$ is not greater than that of a more general rule $X \Rightarrow Z$, it is omitted by the informative rule set. It is clear that rule $Z \Rightarrow XY$ is omitted as well.

Secondly, the informative rule set excludes more rules than those derivable ones. For example, given a small set of transactions: $\{\{1 : X, c_1\}, \{2 : X, c_1\}, \{3 : Y, c_1\}, \{4 : Y, c_1\}, \{5 : X, Y, c_1\}, \{6 : X, Y, c_1\}, \{7 : X, Y, c_1\}, \{8 : X, Y, c_1\}, \{9 : X, Y, c_1\}, \{10 : X, Y, c_2\}\}$. We have the following five rules: $X \Rightarrow c_1(conf = 0.88)$, $Y \Rightarrow c_1(conf = 0.88)$, $XY \Rightarrow c_1(conf = 0.83)$, $X \Rightarrow Y(conf = 0.75)$, and $Y \Rightarrow X(conf = 0.75)$. Rule $XY \Rightarrow c_1(conf = 0.83)$ is omitted by the informative rule set, but not by the non-redundant rule set.

In fact, all derivable rules have something to do with 100% confidence rules, and these rules are not very common in a rule set generated from a transaction database. So, the non-redundant rule set cannot exclude many rules from the association rule set generated from transaction databases.

There is another type of derivable rules, the transitivity rules. For example, if $a \Rightarrow b$ is a 100% confidence association rule and so is $b \Rightarrow c$, then $a \Rightarrow c$ must be a 100% confidence rule and its support is the same as $a \Rightarrow b$. Hence, $a \Rightarrow c$ is derivable. Further, if both $a \Rightarrow b$ and $b \Rightarrow c$ are 100% confidence rules and $c \Rightarrow b$ and $b \Rightarrow a$ have confidence r and s respectively, then rule $c \Rightarrow a$ is derivable. This is because its confidence equals to $s \times t$ and its support is the same as that of $b \Rightarrow a$.

The informative rule set does not exclude these transitive rules while the non-redundant rule set excludes them. However these transitive rules are rare since two consecutive 100% rules are involved. In a rule set generated from a transaction database, there are few transitive rules, so their effect on the size of a rule set can be ignored. For example, in our experiments, there is no such transitive rule generated at all. Hence, in the general case informative rule set is a subset of non-redundant association rule set.

4. The upward closure properties

Most association rule mining algorithms use the upward closure property of infrequent itemsets: if an itemset is infrequent, so are all its super itemsets. Hence, many infrequent itemsets are prevented from being generated in the mining process, and this is the essence of Apriori. If similar properties are applied to the rules omitted by the informative rule set, then we can prevent generation of many rules omitted by the informative rule set. As the result, algorithms based on these properties will be more efficient.

First of all, we present a property that will facilitate our discussion. It is convenient to compare the support of itemsets in order to find subset relationships among their tidsets. This is because we always have support information when mining association rules. We have the following lemma for this purpose.

Lemma 5. $t(X) \subseteq t(Y)$ *if and only if* $sup(X) = sup(XY)$.

Proof: We firstly prove the necessary condition.

Since $t(X) \subseteq t(Y)$, $sup(XY) = |t(XY)|/|D| = |t(X) \cap t(Y)|/|D| = |t(X)|/|D| = sup(X)$.

We then prove the sufficient condition.

Since $sup(X) = sup(XY)$, we have that $|t(X)| = |t(X) \cap t(Y)|$. Hence, the only possibility is $t(X) \subseteq t(Y)$.

This completes the proof. □

We have two upward closure properties for mining the informative rule set. In the following two lemmas, we show they are easy to use in algorithm design but may not be very good in terms of mathematical simplicity.

Lemma 6. *If* $sup(X) = sup(XY)$, *then for any* Z, *rule* $XY \Rightarrow Z$ *and all more specific rules do not occur in the informative rule set.*

Proof: Since $sup(X) = sup(XY)$, we have $t(X) \subseteq t(Y)$. As the result, $XY \Rightarrow Z$ is derivable by Lemma 3, and hence is omitted by the informative rule set.

Furthermore, $t(XX') = t(XX'Y)$ holds for any X'. We have $sup(XX') = sup(XX'Y)$. Similarly, rule $XX'Y \Rightarrow Z$ is omitted by the informative rule set.

Consequently, rule $XY \Rightarrow Z$ and all other more specific rules are omitted by the informative rule set. □

It is clear that this lemma is for those derivable rules defined by Lemma 3.

Lemma 7. *If* $sup(X \neg Z) = sup(XY \neg Z)$, *then rule* $XY \Rightarrow Z$ *and all more specific rules do not occur in the informative rule set.*

Proof: Since $sup(X \neg Z) = sup(XY \neg Z) = sup(X \neg ZY \neg Z)$, we have $t(X \neg Z) \subseteq t(Y \neg Z)$. As the result, $XY \Rightarrow Z$ cannot be included in the informative rule set by Lemma 2.

Furthermore, $t(XX' \neg Z) = t(XX'Y \neg Z)$ holds for any X'. We have $sup(XX' \neg Z) = sup(XX'Y \neg Z)$. Similarly, rule $XX'Y \Rightarrow Z$ cannot be included in the informative rule set.

Consequently, rule $XY \Rightarrow Z$ and all rules that are more specific must be omitted by the informative rule set. □

Clearly, this lemma is for those rules defined by Lemma 2.

Finally, we discuss the relationship between the two lemmas. If $sup(X) = sup(Xz)$, then $sup(X \neg Y) = sup(Xz \neg Y)$ for all Y. However, the reverse relationship does not hold. Hence, Lemma 7 is more general than Lemma 6 and we can omit more rules by Lemma 7 than by Lemma 6. Lemma 6 is actually for derivable rules, which are a part of rules omitted by the informative rule set.

These two lemmas enable us to prune unwanted rules in a "forward" fashion before they are actually generated. In fact we can prune a set of rules when we prune each rule not in the informative rule set in the early stages of the computation. This allows us to construct efficient algorithms to generate the informative rule set.

5. Mining algorithm

5.1. Basic idea and storage structure

We propose a direct algorithm to mine the informative rule set. Instead of first finding all frequent itemsets and then forming rules, the proposed algorithm generates informative rule set directly. An advantage of doing so is that it avoids generating many frequent itemsets that lead to rules omitted by the informative rule set.

The proposed algorithm is a level-wise algorithm, which searches for rules from antecedent of 1-itemset to antecedent of l-itemset level by level. In each level, we select qualified rules, which could be included in the informative rule set, and prune those unqualified rules. The efficiency of the proposed algorithm is based on the fact that a number of rules omitted by the informative rule set are prevented from being generated once a more general rule is pruned by Lemma 6 or 7. Consequently, the searching space is reduced after each level's pruning. The number of phases of accessing a database is bounded by the length of the longest rule in the informative rule set plus one.

In the proposed algorithm, we extend a set enumeration tree (Rymon, 1992) as the storage structure, called *candidate tree*. A simplified candidate tree is illustrated in figure 1. The tree in figure 1 is completely expanded, but in practice only a small part may need to be expanded. We note that each set in the tree is unique and is used to identify the node, called *identity set*. We also note that labels are locally distinct to each other under the same parent node in a layer, and labels along a path from the root to the node form exactly the identity set of the node. This is very convenient for retrieving the itemset and counting its frequency. In our algorithm a node is used to store a set of rule candidates.

5.2. The algorithm

The set of all items is used to build a candidate tree. A node in the candidate tree stores two sets $\{A, Z\}$, where A is an itemset, the identity set of the node, and Z is a subset of the

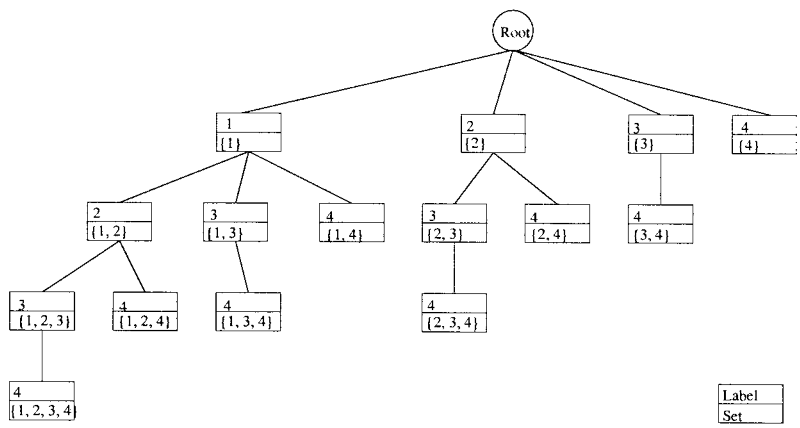

Figure 1. A fully expanded candidate tree over the set of items {1, 2, 3, 4}.

identity itemset, called potential target set in which each item can be the consequence of an association rule. For example, $\{\{abc\}, \{ab\}\}$ is a set of candidates of two rules, namely, $bc \Rightarrow a$ and $ac \Rightarrow b$. It is clear that the potential target set is initialized by the itemset itself. When there is a case satisfying Lemma 7, for example, $sup(a \neg c) = sup(ab \neg c)$, then we remove c from the potential target set, and accordingly all rules such as $abX \rightarrow c$ cannot be generated afterwards.

We first illustrate how to generate a new candidate node. For example, we have two sibling nodes $\{\{abc\}, \{ab\}\}$ and $\{\{abd\}, \{ad\}\}$, then the new candidate is $\{\{abcd\}, \{ad\}\}$, where $\{ad\} = (\{ab\} \cup \{d\}) \cap (\{ad\} \cup \{c\})$. Hence the only two candidate rules that could be included in the informative rule set in this case are $bcd \Rightarrow a$ and $abc \Rightarrow d$ given that $abcd$ is frequent. Item c is omitted for the target set of $\{\{abc\}, \{ab\}\}$, and this means that $ab \Rightarrow c$ and all more specific rules, such as $abd \Rightarrow c$ will not occur in the informative rule set. So item c does not appear in the target set of $\{\{abcd\}, \{ad\}\}$. The same reason for omitting item a. We use $\{ab\} \cup \{d\}$ because d is new to set abc, and we do not want to miss rule $abc \Rightarrow d$, and the same reason for using $\{ad\} \cup \{c\}$.

We then show how to remove unqualified candidates. One way is by the frequency requirement. For example, if $sup(abcd) < \sigma$, then we remove the node whose identity set is $abcd$, called node $abcd$. Note that here a node in the candidate tree contains a set of candidate rules. Another method is by the properties of the informative rule set, which consists of two steps. First, for candidate node $\{A^l, Z\}$ and an item $z \in Z$, where A^l stands for an l-itemset, if there is $sup((A^l \backslash z) \neg z) = sup((A^{l-1} \backslash z) \neg z)$ for $(A^l \backslash z) \supset (A^{l-1} \backslash z)$, then remove the z from Z by Lemma 7. Secondly, we say node $\{A^l, Z\}$ is *restricted* if there is $sup(A^l) = sup(A^{l-1})$ for $A^l \supset A^{l-1}$. A restricted node does not extend its potential target set and keeps it as that of node $\{A^{l-1}, Z\}$. The reason is that all rules $A^{l-1}X \Rightarrow c$ for any X and c are omitted from the informative rule set by Lemma 6, so we need not generate such candidates. This potential target set is removable by Lemma 7, and a restricted node is *dead* when its potential target set is empty. All supersets of the itemset of a dead node are unqualified candidates, so we need not generate them.

We give the top level of the informative rule mining algorithm as the following.

Algorithm: Informative rule set miner

Input: Database D, the minimum support σ and the minimum confidence ψ.
Output: The informative rule set R.

(1) Set the informative rule set $R = \emptyset$
(2) Count support of 1-itemsets
(3) Initialize candidate tree T
(4) Generate new candidates as leaves of T
(5) While (new candidate set is non-empty)
(6) Count support of the new candidates
(7) Prune the new candidate set
(8) Include qualified rules from T to R
(9) Generate new candidates as leaves of T
(10) Return rule set R

The first 3 lines are general description that are self-explanatory. We will elaborate the two functions, Candidate generator in line 4 and 9 and Pruning in line 6. They are listed as follows.

First of all, we introduce some notations in the functions: n_i is a candidate node in the candidate tree, labeled by an item (vertex) i_{n_i}, contains an identity itemset A_{n_i} and a potential target set Z_{n_i}; T_l is the l-th level of candidate tree; $\mathcal{P}^l(A)$ is the set of all l-subsets of A; n_A is a node whose identity itemset is A. All items are in the lexicographic order.

Function Rule candidate generator

(1) for each node $n_i \in T_l$
(2) for each sibling node n_j $(i_{n_j} > i_{n_i})$
(3) generate a new candidate node n_k as a son of n_i such that
//Combining
(4) $A_{n_k} = A_{n_i} \cup A_{n_j}$
(5) $Z_{n_k} = (Z_{n_i} \cup i_{n_j}) \cap (Z_{n_j} \cup i_{n_i})$
//Pruning
(6) if $\exists A \in \mathcal{P}^l(A_{n_k})$ but $n_A \notin T_l$ then remove n_k
(7) else if n_A is restricted then mark n_k restricted and let $Z_{n_k} = Z_{n_A} \cap Z_{n_k}$
(8) else $Z_{n_k} = (Z_{n_A} \cup (A_{n_k} \backslash A)) \cap Z_{n_k}$
(9) if n_k is restricted and $Z_{n_k} = \emptyset$, remove node n_k

We generate the $(l+1)$-layer candidates from the l layer nodes. Firstly, we combine a pair of sibling nodes and insert their combination as a new node in the next layer. Secondly, if any of its l-sub itemset cannot get enough support then we remove the node. If an item is not qualified to be the target of a rule included in the informative rule set, then we remove the target from the potential target set.

Note that in line 6, not only a superset of an infrequent itemset is removed, but also a superset of a frequent itemset of a dead node is removed. The former is common in association rule mining, and the latter is unique for the informative rule mining. A dead node is removed in line 9. As the result, informative rule mining doesn't generate all frequent itemsets.

Function Pruning

(1) for each $n_i \in T_{l+1}$
(2) if $sup(A_{n_i}) < \sigma$, remove node n_i and return
(3) if n_i is not restricted node, do
(4) if $\exists n_j \in T_l$ for $A_{n_j} \subset A_{n_i}$ such that $sup(A_{n_j}) = sup(A_{n_i})$
then mark n_i restricted and let $Z_{n_i} = Z_{n_i} \cap Z_{n_j}$ // Lemma 6
(5) for each $z \in Z_{n_i}$
(6) if $\exists n_j \in T_l$ for $(A_{n_j} \backslash z) \subset (A_{n_i} \backslash z)$ such that
$sup((A_{n_j} \backslash z) \cup \neg z) = sup((A_{n_i} \backslash z) \cup \neg z)$
then $Z_i = Z_i \setminus z$. // Lemma 7
(7) if n_i is restricted and $Z_{n_i} = \emptyset$, remove node n_i

We prune a rule candidate from two aspects, frequency requirement for association rules and qualification requirement for the informative rule set. The method for pruning infrequent rules is the same as that for association rule mining. As for the method for pruning unqualified candidates for the informative rule set, we restrict the possible targets in the potential target set of a node (a possible target is equivalent to a rule candidate) and remove a restricted node when its potential target set is empty.

5.3. Correctness and efficiency

Lemma 8. *The algorithm generates the informative rule set properly.*

Proof: We will prove the claim from two aspects. One is that the candidate tree can generate all single consequence association rules directly, and the other is that the pruned rules are those which must be omitted by the informative rule set.

Basically a candidate tree can enumerate all subsets of the set of all items, and store every itemset in a node of the tree as the identity set of the node. The itemset stored in a child node is a superset of the itemset stored in its parent node, so a set of super itemsets are stored in a branch of the tree. Once we have removed those infrequent branches, all remaining nodes store frequent itemsets. Let the potential target set of an itemset be the itemset itself. We can then obtain all single consequence association rules directly.

Now, we will prove that all pruned candidate rules are those which must be omitted by the informative rule set from three aspects.

Firstly, in our algorithm, the potential target set is a subset of the itemset stored in a node, $Z \subseteq A$, and some items are omitted from set Z by Lemma 7. Specifically, if $sup(A \neg z) = sup(A' \neg z)$ for $A' \supset A$ then all rules $A'' \Rightarrow z$ for $A'' \supseteq A'$ are omitted from the informative rule set. Hence, we can remove all rule candidates $A'' \Rightarrow z$, and equivalently, remove z from every potential target set of every node in the subtree rooted by node $n_{A'}$ in the algorithm.

Secondly, for all restricted nodes, we do not extend their potential target sets while expanding their itemsets. Since if $sup(A) = sup(A')$ for $A' \supset A$ then all rules $A'' \Rightarrow c$ for $A'' \supseteq A'$ and any c are omitted from the informative rule set by Lemma 6. Given a restricted node A'' where $A'' \supset A$ and $sup(A'') = sup(A)$, all rules $A''\backslash z \Rightarrow z$ where $z \in \{A''\backslash A\}$ must be omitted from the informative set. The potential target set Z'' for node $n_{A''}$ must be a subset of A, and hence Z'' doesn't need to be extended.

Finally, we do not generate a candidate node that stores a superset of the identity set of a dead node. We know that the potential target set of a restricted node A' is only a subset of A where A is the smallest subset of A' such that $sup(A) = sup(A')$. If all items in Z are not qualified consequences of A', then A' and all its supersets cannot contain rules to be included in the informative rule set.

In summary, the algorithm generates informative rule set properly. □

It is very hard to give a closed form of efficiency of the algorithm. However, we expect improvements over other association rule mining algorithms based on the following reasons. Firstly, our algorithm does not generate all frequent itemsets, because some frequent itemsets cannot contain rules being included in the informative rule set. Secondly, our algorithm does not test all possible rules in each generated frequent itemset because some items in an itemset are not qualified as consequences for rules being included in the informative rule set.

The phases of accessing a database is bounded by the length of the longest rule in the informative rule set plus one.

6. Discussion

In this section we will present some discussions on why we choose the confidence priority model for prediction. Apparently, it extends a classification model by allowing a set of items to be prediction output. Reasons for using confidence priority are listed as follows.

First, since confidence is the accuracy of a rule based on the data from which it is generated, naturally we prefer the highest accuracy rules.

Secondly, confidence approximates to the true accuracy in a large database. Predictions are usually made on the data that is different sample from the data where rules are generated. We call the test data and training data respectively. Clearly, confidence is the training accuracy, and the true accuracy is the test accuracy on a large-size sample. However, we never know the true accuracy in the rule generation stage and have to estimate it. Here is an estimation of the true accuracy (Mitchell, 1997).

$$acc(A \Rightarrow c) = conf(A \Rightarrow c) \pm z_N \sqrt{\frac{conf(A \Rightarrow c)(1 - conf(A \Rightarrow c))}{|cov(A \Rightarrow c)|}}$$

where z_N is a constant related with a statistical confidence interval and $|cov(A \Rightarrow c)|$ is the number of transactions containing A. In a large database, $|cov(A \Rightarrow c)|$ is usually big. Hence confidence approximates to the true accuracy.

Thirdly, the predictions provided by the confidence priority will not be significantly affected by the changing of minimum confidence. In the confidence priority model, each prediction is made by a rule with the maximum confidence, and hence the distance to the minimum confidence is also maximized. As a result, the change of the minimum confidence would not significantly affect the prediction.

Alternatively, we may have a support priority model. The support priority model is the one that selects the matching rule with maximum support to make prediction each time. It reflects the emphasis on the popularity of a prediction.

Consider the prediction from the support priority model as a sequence of items. For any input itemset, the prediction sequence from the informative association rule set is identical to the prediction sequence from the association rule set. This is because all highest support rules are contained in the informative association rule set. In fact, to generate the same prediction sequence as the association rule set, the support priority model only needs a subset of informative rule set. Though the rule set is small, it loses the highest confidence information which is crucial in predictions.

Consequently, we choose to use the confidence priority model in this paper. The resulting informative rule set contains highest confidence information as well as highest support information, therefore it suits for various applications.

7. Experimental results

In this section, we show that the informative rule set is much smaller than both the association rule set and non-redundant association rule set. We further show that it can be generated more efficiently with fewer interactions with a database. Finally, we show the efficiency improvement gain from the fact that the proposed algorithm for the informative rule set accesses the database fewer times and generates fewer candidates than Apriori for the association rule set.

Since the informative rule set contains only single-target rules, for fair comparisons, we assume that the association rule set and non-redundant rule set in this section contain only single-target rules as well. The reason for the comparison with the non-redundant rule set is that it can also make the same predictions as the association rule set.

The two test transaction databases, T10.I6.D100K.N2K and T20.I6.D100K.N2K, are generated by the synthetic data generator from QUEST of IBM Almaden research center. Both databases contain 1000 items and 100,000 transactions. We chose the minimum support in the range such that 70% to 80% of all items are frequent, and fix the minimum confidence to 0.5.

Sizes of different rule sets are listed in figure 2. It is clear that the informative rule set is much smaller than both the association rule set and non-redundant rule set. The size difference between informative rule set and association rule set becomes more evident when the minimum support decreases, as does the size difference between informative rule set and non-redundant rule set. This is because the length of rules becomes longer when the minimum support decreases, and long rules are more likely to be omitted by the informative rule set than short rules. From our discussion in Section 4, we know that all redundant rules are connected with at least one 100% confidence rule. However, in these randomly generated databases, there are not many 100% confidence rules. Hence there is little difference in size between association rule set and non-redundant rule set. As the result, in the following comparisons, we only compare the informative rule set with the association rule set.

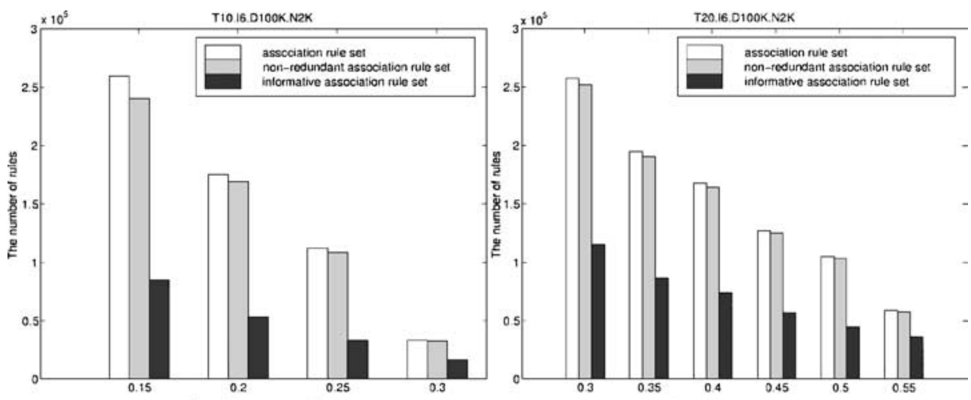

Figure 2. Sizes of different rule sets.

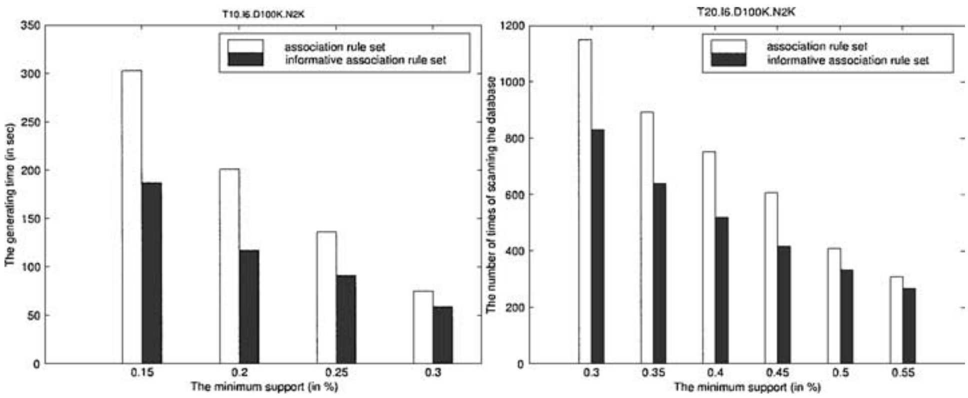

Figure 3. Generating time for different rule sets.

Next, we shall compare the efficiencies for generating informative rule set and association rule set. We implemented Apriori on the same data structure as the proposed algorithm and generated only single-target association rules. Our experiments were conducted on a Sun server with two 200 MHz UltraSPARC CPUs.

The times for generating association rule set and informative rule set are listed in the figure 3. We can see that the proposed algorithm for mining informative rule set is more efficient than Apriori for mining single-target association rule set. This is because the proposed algorithm does not generate all frequent itemsets, and does not test all items as targets in a frequent itemset. The improvement of efficiency becomes more evident when the minimum support decreases. This is consistent with the deduction of rules being omitted from an association rule set as shown in figure 2.

Furthermore, the number of times of accessing database required by proposed algorithm is smaller than Apriori, as showed in figure 4. This is because the proposed algorithm avoids

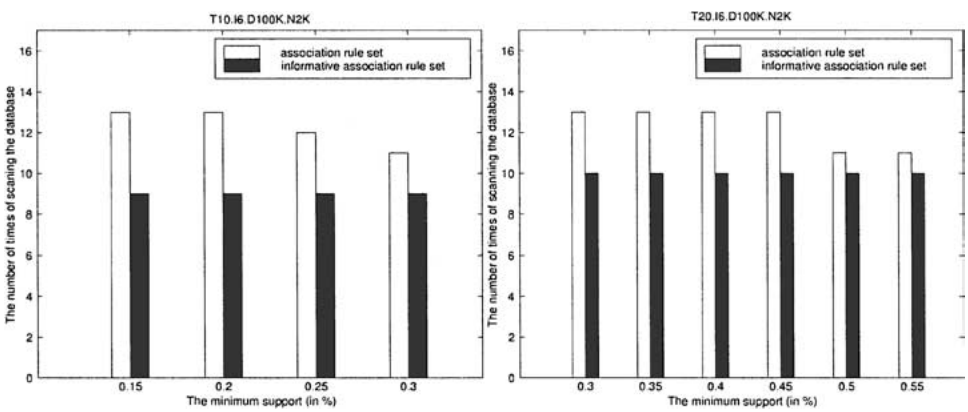

Figure 4. The number of times of accessing the database.

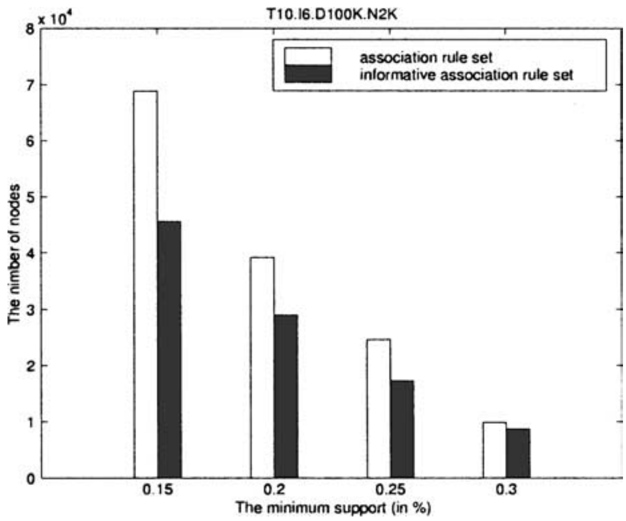

Figure 5. The number of candidate nodes.

generating many long frequent itemsets that contain no rules included in an informative rule set. From the results, we also know that long rules are easier to be omitted by informative rule set than short rules. Clearly, this number is clearly much smaller than the number of frequent itemsets which are needed to access a database in direct association rule generating algorithms.

To better understand efficiency improvement of the proposed algorithm over Apriori, we list the number of nodes in a candidate tree for both association and informative rule sets in figure 5. The numbers are all frequent itemsets for Apriori to generate all association rules and partial frequent itemsets for the proposed algorithm to generate an informative association rule set. We can see that in mining informative rule set, the number of itemsets searched is smaller than that of all frequent itemsets for forming all association rules. This is the reason for efficiency improvement and reduction in the number of times to access a database. This result also indicates that the proposed algorithm uses less memory space than Apriori does.

The improvement is very significant since the proposed algorithm is faster and uses less memory in comparison with Apriori. Especially, the noticeable improvement occurs at small support, which is the bottleneck of association rule mining. In the worse case, e.g. when support is big, the proposed algorithm accesses a database the same times as Apriori does. Both Apriori and our proposed algorithm are level-wise (breadth first) algorithms, and they access a database much less often than non-redundant rule set generation (Zaki, 2000) that is a depth first algorithm.

8. Conclusion

We have defined a new rule set, informative rule set, that generates the same prediction sequences as association rule set according to the confidence priority. The size of informative rule set is significantly smaller than association rule set, especially when the minimum support is small. We have studied the relationships between informative rule set and non-redundant association rule set, and revealed that informative rule set is a subset of non-redundant association rule set. We have also studied the upward closure properties of informative rule set for omission of unnecessary rules from the set, and presented a direct algorithm to efficiently mine the informative rule set without generating all frequent itemsets first. The experimental results have confirmed that our informative rule set is significantly smaller and can be generated more efficiently than both association rule set and non-redundant association rule set. The experimental results have also shown that this efficiency improvement is resulted from the fact that generation of informative rule set needs fewer candidates and database accesses than that of association rule set. The number of database accesses of the proposed algorithm is much smaller than other direct methods for generating association rules on all items.

We notice that a predictive rule set can be very small by incorporating some domain knowledge, and the significance of this work is that such small predictive rule set can be derived directly from the informative rule set instead of the association rule set. By doing this, much time can be saved. This is because informative association rule set can be generated more efficiently, and pruning on a smaller rule set is more efficient than pruning on a larger rule set.

Although the informative rule set provides the same prediction sequence as the association rule set, there may exist other definitions of "interestingness" in different applications. How to further incorporate informative rule set generation with different criteria remains a subject of future work.

Appendix

Before proving Lemma 4, we introduce some terms used in Zaki (2000). For tidset Y, let mapping $i(Y)$ be the maximum itemset that is contained in all transactions in Y. Let $c_{it}(X)$ denote the composition of two mappings $i \circ t(X) = i(t(X))$, and $c_{ti}(Y) = t \circ i(X) = t(i(Y))$. Itemset X is closed if $X = c_{it}(X)$. The support of an itemset equals that of its closed itemset.

We first restate two theorems in paper (Zakik, 2000).

Theorem 5. *Let R_i stand for a* 100% *confidence rule $X^i \Rightarrow Y^i$, $\mathcal{R} = \{R_1, \ldots, R_n\}$ be a set of rules such that $I_1 = c_{it}(X^i \cup Y^i)$, and $I_2 = c_{it}(Y^i)$ for all rules R_i. Then all the rules are equivalent to the* 100% *confidence rule $I_1 \Rightarrow I_2$. Further, all rules other than the most general ones are redundant.*

Theorem 6. *Let R_i stand for a rule $X^i \Rightarrow Y^i$ with confidence less than* 100%, *and let $\mathcal{R} = \{R_1, \ldots, R_n\}$ be a set of rules such that $I_1 = c_{it}(X^i)$, and $I_2 = c_{it}(X^i \cup Y^i)$ for all rules R_i. Then all the rules are equivalent to rule $I_1 \Rightarrow I_2$. Further, all rules other than the most general ones are redundant.*

The following lemma needs to be proven:

Lemma 4. *Redundant rules given in Theorems* 5 *and* 6 (*Zaki*, 2000) *are derivable rules.*

Proof: For convenience, we omit the upper script of X and Y. We notice that if $I = c_{it}(X)$ then both $I \supseteq X$ and $t(I) = t(X)$ hold. We will use this relationship throughout the proof.

First, let us look at Theorem 5. Suppose that $X \Rightarrow Y$ is one of the most general rules in the rule set $\mathcal{R}$. Since $X \Rightarrow Y$ is a 100% confidence rule, we have $t(X) \subseteq t(Y)$. Let $XZ \Rightarrow I_2$ be an equivalent rule of $I_1 \Rightarrow I_2$ and $Z \neq \emptyset$. From the condition given in Theorem 5, we have $t(XZ) = t(XY) = t(X) \cap t(Y) = t(X)$. Hence we obtain $t(X) \subseteq t(Z)$. As the result, rule $XZ \Rightarrow I_2$ is derivable by Lemma 3. Let $I_1 \Rightarrow YZ'$ be another equivalent rule of $I_1 \Rightarrow I_2$ and $Z' \neq \emptyset$. Since $t(YZ') = t(Y)$, we have $t(Y) \subseteq t(Z')$. As the result, $I_1 \Rightarrow YZ'$ is derivable by Lemma 3. Hence, we can conclude that all equivalent rules of $I_1 \Rightarrow I_2$ other than the most general ones given in Theorem 5 are derivable.

Next, let us look at Theorem 6. Suppose that $X \Rightarrow Y$ is one of the most general rules in the rule set $\mathcal{R}$. Let $XZ \Rightarrow I_2$ be an equivalent rule of $I_1 \Rightarrow I_2$ and $Z \neq \emptyset$. Since $t(XZ) = t(X)$, we have $t(X) \subseteq t(Z)$. Hence, rule $XZ \Rightarrow I_2$ is derivable by Lemma 3. Let $I_1 \Rightarrow XYZ'$ be another equivalent rule of $I_1 \Rightarrow I_2$ and $Z' \neq \emptyset$. From the condition given by Theorem 6, we have $t(XYZ') = t(XY)$. Hence, We obtain $t(XY) \subseteq t(Z')$. As the result, rule $I_1 \Rightarrow XY$ is derivable by Lemma 3. Furthermore, $I_1 \Rightarrow XY$ can be derived from $X \Rightarrow XY$, or equivalently from $X \Rightarrow Y$. Hence, we can conclude that all equivalent rules of $I_1 \Rightarrow I_2$ other than the most general ones given in Theorem 6 are derivable. $\square$

Acknowledgments

This work was partially supported by a research grant funded by Japan Society for the Promotion of Science (JSPS) under its General Research Scheme **B** Grant No.14380139.

References

Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining Associations Between Sets of Items in Massive Databases. In *Proc. of the ACM SIGMOD Int'l Conference on Management of Data* (pp. 207–216).

Agrawal, R. and Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the Twentieth International Conference on Very Large Databases* (pp. 487–499). Santiago, Chile.

Bayardo, R. and Agrawal, R. (1999). Mining the Most Interesting Rules. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 145–154). N.Y.: ACM Press.

Bayardo, R., Agrawal, R., and Gunopulos, D. (1999). Constraint-Based Rule Mining in Large, Dense Database. In *Proc. of the 15th Int'l Conf. on Data Engineering* (pp. 188–197).

Han, J., Pei, J., and Yin, Y. (2000). Mining Frequent Patterns Without Candidate Generation. In *Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00)* (pp. 1–12).

Holsheimer, M., Kersten, M., Mannila, H., and Toivonen, H. (1995). A Perspective on Databases and Data Mining. In *1st Intl. Conf. Knowledge Discovery and Data Mining* (p. 10).

Houtsma, M. and Swami, A. (1995). Set-Oriented Mining for Association Rules in Relational Databases. In *Proceedings of the 11th International Conference on Data Engineering* (pp. 25–34). Los Alamitos, CA, USA: IEEE Computer Society Press.

Liu, B., Hsu, W., and Ma, Y. (1999). Pruning and Summarizing the Discovered Associations. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining (SIGKDD 99).*

Mannila, H., Toivonen, H., and Verkamo, I. (1994). Efficient Algorithms for Discovering Association Rules. In *AAAI Wkshp. Knowledge Discovery in Databases* (pp. 181–192). AAAI Press.

Mitchell, T.M. (1997). *Machine Learning*. McGraw-Hill.

Ng, R., Lakshmanan, L., Han, J., and Pang, A. (1998). Exploratory Mining and Pruning Optimizations of Constrained Associations Rules. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD-98)* (pp. 13–24). ACM SIGMOD Record 27(2), New York: ACM Press.

Park, J.S., Chen, M., and Yu, P.S. (1995). An Effective Hash Based Algorithm for Mining Association Rules. In *ACM SIGMOD Intl. Conf. Management of Data*.

Rymon, R. (1992). Search Through Systematic Set Enumeration. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning* (pp. 539–552) Cambridge, MA: Morgan Kaufmann.

Savasere, A., Omiecinski, R., and Navathe, S. (1995). An Efficient Algorithm for Mining Association Rules in Large Databases. In *Proceedings of 21th International Conference on Very Large Data Bases (VLDB95)* (pp. 432–444).

Shenoy, P., Haritsa, J.R., Sudarshan, S., Bhalotia, G., Bawa, M., and Shah, D. (1999). Turbo-Charging Vertical Mining of Large Databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD-99)* (pp. 22–33). ACM SIGMOD Record 29(2), Dallas, Texas: ACM Press.

Toivonen, H., Klemettinen, M., RonKainen, P., Hatonen, K., and Mannila, H. (1995). Pruning and Grouping Discovered Association Rules. In *Workshop Notes of the ECML-95 Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases* (pp. 47–52).

Webb, G.I. (2000). Efficient Search for Association Rules. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-00)*, (pp. 99–107). N.Y.: ACM Press.

Zaki, M.J. (2000). Generating Non-Redundant Association Rules. In *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 34–43).

Zaki, M.J., Parthasarathy, S., Ogihara, M., and Li, W. (1997). New Algorithms for Fast Discovery of Association Rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)* (p. 283). AAAI Press.