Mining the smallest association rule set for predictions

Jiuyong Li, Hong Shen and Rodney Topor
School of Computing and Information Technology
Griffith University
Brisbane, Australia, QLQ 4111
{jiuyong, hong, rwt}@cit.gu.edu.au

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 subset the 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 present an algorithm to directly generate the informative rule set, i.e., without generating all frequent itemsets first, and that accesses the database less often than other unconstrained direct methods. We show experimentally that the informative rule set is much smaller than both the association rule set and the non-redundant association rule set, and that it can be generated more efficiently.

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 to describe 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 subfield of data mining and has wide application in many fields. Two key problems with association rule mining are the high cost of generating association rules and the large number of rules that are normally generated. Much work has been done to address the first problem. Methods for reducing the number of rules generated depend on the application, 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 particular customer, we wish to predict what other purchases will be made by that customer.

The association rule set [1] can be used for prediction if the high cost of finding and applying the rule set is not a concern. The constrained and optimality association sets [4, 3] 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 [17] can be used, after some extension, but can be large as well.

We propose the use of a particular rule set, called the informative (association) rule set, that is smaller than the association rule set and that makes the same predictions under natural assumptions described below.

The general method of generating association rules by first generating frequent itemsets can be unnecessarily expensive, as many frequent itemsets do not lead to useful association rules. We present a direct method for generating the informative rule set that does not involve generating the frequent itemsets first. Unlike other algorithms that generate rules directly, our method does not constrain the consequences of generated rules as in [3, 4] and accesses the database less often than other unconstrained methods [16].

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

2 Related work

Association rule mining was first studied in [1]. Most research work has been on how to mine frequent itemsets efficiently. Apriori [2] is a widely accepted approach, and there have been many enhancements to it [6, 7, 9, 11, 13]. In addition, other approaches have been proposed [5, 14, 18], mainly by using more memory to save time. For example, the algorithm presented in [5] organizes a database into a condensed structure to avoid repeated database ac-

cesses, and algorithms in [14, 18] use the vertical layout of databases.

Some direct algorithms for generating association rules without generating frequent itemsets first have previously been proposed [4, 3, 16]. Algorithms presented in [4, 3] focused only on one fixed consequence and hence is not efficient for mining all association rules. The algorithm presented in [16] needs to scan a database as many times as the number of all possible antecedents of rules. As a result, it may not be efficient when a database cannot be retained in the memory.

There are also two types of algorithms to simplify the association rule set, direct and indirect. Most indirect algorithms simplify the set by post-pruning and reorganization, as in [15, 8, 10], which can obtain an association rule set as simple as a user would like but does not improve efficiency of the rule mining process. There are some attempts to simplify the association rule set directly. The algorithm for mining constraint rule sets is one such attempt [4]. It produces a small rule set and improves mining efficiency since it prunes unwanted rules in the processing of rule mining. However, a constraint rule set contains only rules with some specific items as consequences, as do the optimality rule sets [3]. 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 [17]. However, the non-redundant rule set is still too large for prediction.

3 The informative rule set

3.1 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) is the ratio of the number of transactions containing the itemset to the number of all transactions in a database, denoted by $sup(X)$. 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. $sup(X \cup Y)/sup(X)$ 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 $sup(X) - sup(XZ)$.

3.2 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 matched rule with the highest conference from the rule set, and then put the consequence of the rule into 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. Repeat selecting the next highest confidence matched rule from the remaining rule set until the user is satisfied or there is not rule to select.

We have noticed that some rules in the association rule set will never been 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 will predict exactly the same as the association rule set, the same set of prediction items in the same generated order. Here, we consider the order because a user may stop selection at any time, and we will guarantee to obtain the same prediction items in this case.

Formally, given an association rule set R and an itemset P, we say that the *predictions* for P from R is a sequence of items Q. The sequence of 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 consequent of r is added to Q. After adding a consequence to Q, all rules whose consequences are in Q are removed from R.

To exclude those rules that 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.*

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 threshholds). 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. 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. 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 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 the 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 for* P *from* R_I *equals the prediction sequence for* 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 omitted 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 do not affect prediction from R_A for any P.

Consider a single-target rule r' omitted by R_I, there must be another rule r in both R_I and R_A such that the $r \subset r'$ and $conf(r) \geq conf(r')$. When r' matches P, r does. 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 sequenc. So, omitting r' does not affect prediction from R_A.

Consider 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 both R_I and R_A for $A' \subseteq A$ 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 will prove the minimum property. Suppose we omit one rule $X \Rightarrow c$ from the 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 not 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 $X \Rightarrow c$. As a result, the two prediction sequences would 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.

Consequently, the theorem is proved. □

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), $t(X) = t(XZ) \cup t(X\neg Z)$. 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 excluded by the informative rule set. □

This is an important property for 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 the next section.

4 Upward closed properties for generating informative rule sets

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

First of all, we discuss a property that will facilitate the following discussions. It is convenient to compare 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 a relationship for this purpose.

Lemma 3 *$t(X) \subseteq t(Y)$ if and only if $sup(X) = sup(XY)$.*

We have two upward closed properties for mining the informative association rule set. In the following two lemmas, we use a description that is easy to use in algorithm design but may not be very good in terms of mathematical simplicity

As a direct result of Lemma 2 and 3, we have

Lemma 4 *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.*

The following special case is useful in practice.

Lemma 5 *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.*

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 Algorithm

5.1 Basic idea and storage structure

We proposed 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 is that it avoids generating many frequent itemsets that produce rules excluded 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 every 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 excluded by the informative rule set are prevented from being generated once a more general rule is pruned by Lemma 4 or 5. Consequently, searching space is reduced after every level's pruning. The number of phases of scanning a database is bounded by the length of the longest rule in the informative rule set.

In the proposed algorithm, we extend a set enumeration tree [12] 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 is expanded. We note that every set in the tree is unique and is used to identified the node, called *identity set*. We also note that labels are locally distinct with each other under a 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 Algorithm for mining the informative rule set

The set of all items is used to build a candidate tree. A node in the candidate tree stores two sets $\{A, Z\}$. A is an itemset, the identity set of the node, and Z is a subset of the identity itemset, called potential target set where every 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 4, for example, $sup(a\neg c) = sup(ab\neg c)$, then

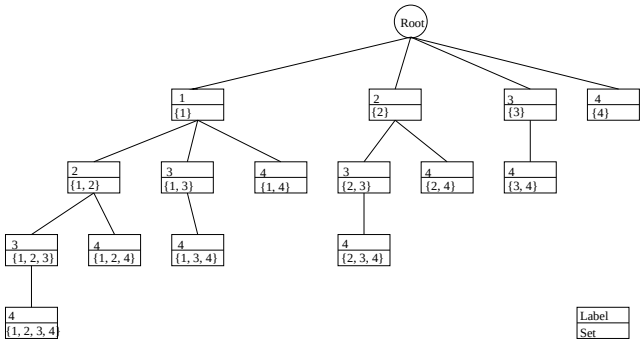

Figure 1. A fully expanded candidate tree over the set of items $\{1, 2, 3, 4\}$

we remove c from the potential target set, and accordingly all rules such as $abX \rightarrow c$ cannot be generated afterwards.

We firstly 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 $abcd$ is frequent.

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 $abcd$, simply called node $abcd$. Please note here that a node in the candidate tree contains a set of candidate rules. Another method is by the properties of the informative rule set, and again consists of two cases. Firstly, given a candidate node $\{A^l, Z\}$ where A^l means that A^l is a l-itemset. For an item $z \in Z$, when 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 4. Secondly, we say node $\{A^l, Z\}$ is *restricted* when 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 excluded from the informative rule set by Lemma 5, so we need not generate such candidates. This potential target set is removable by Lemma 4, and a restricted node is *dead* when its potential target set is empty. All super sets 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: The informative rule 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, and we do not explain them here. We will emphasize on two functions, Candidate generator in line 4 and 9 and Pruning in line 6. They are listed as follows.

We begin with introducing notations in the functions. n_i is a candidate node in the candidate tree. It is labeled by an item i_{n_i} and it consists of 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. The set of items are ordered lexically.

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.

Please note that in line 6, not only a super set of an infrequent itemset is removed, but also a super set of a frequent itemset of a dead node is removed. The former case is common in association rule mining, and the latter case is unique for the informative rule mining. A dead node is removed in line 9. Accordingly, in the informative rule

mining, we need not to 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 4
(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 5
(7) if n_i is restricted and $Z_{n_i} = \emptyset$, remove node n_i

We prune a rule candidate from two aspects, the frequency requirement for association rules and the qualification requirement for the informative rule set. The method for pruning infrequent rules is the same as that of a general association rule mining algorithm. As for the method in 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 6 *The algorithm generates the informative rule set properly.*

It is very hard to give a closed form of efficiency for the algorithm. However, we expect certain improvements over other association rule mining algorithms based on the following reasons. Firstly, it does not generate all frequent itemsets, because some frequent itemsets cannot contain rules being included in the informative rule set. Secondly, it does not test all possible rules in every 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 scanning a database is bounded by the length of longest rule in the informative rule set.

6 Experimental results

In this section, we show that the informative rule set is much smaller than both the association rule set and the non-redundant association rule set. We further show that it can be generated more efficiently with less number of interactions with a database. Finally, we show that the efficiency improvement gains 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 a fair comparison, the association rule set and the 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 the non-redundant rule set can make the same predictions the association rule set.

The two testing 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 have 1000 items and contain 100,000 transactions. We chose the minimum support in the range such that 70% to 80% of all items are frequent, and fixed the minimum confidence as 0.5.

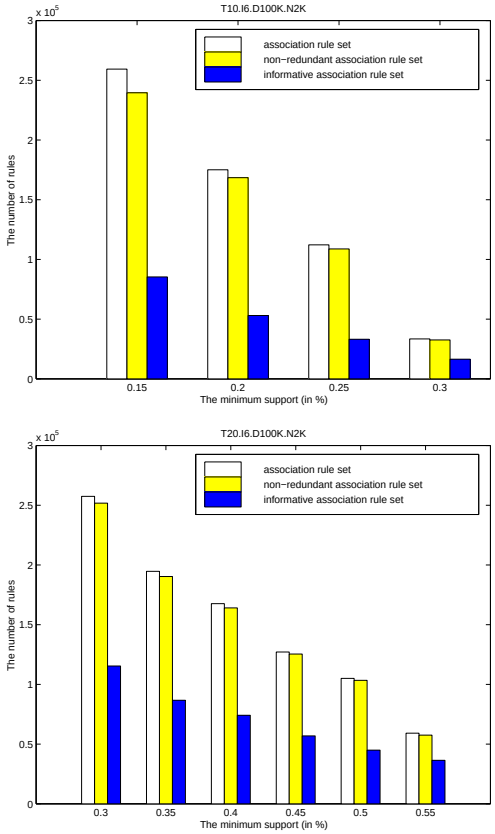

Figure 2. Sizes of different rule sets

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 the non-redundant rule set. The size difference between an informative rule set and an association rule set becomes more evident when the minimum support decreases, and as does the size difference between an informative rule set and a non-redundant rule set. This is because the length of rules becomes longer when the mini-

mum support decreases, and long rules are more likely to be excluded by the informative rule set than short rules. There is little difference in size between an association rule set and a non-redundant rule set. So, in the following comparisons, we only compare the informative rule set with the association rule set.

Now, we will compare generating efficiency of the informative rule set and the 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.

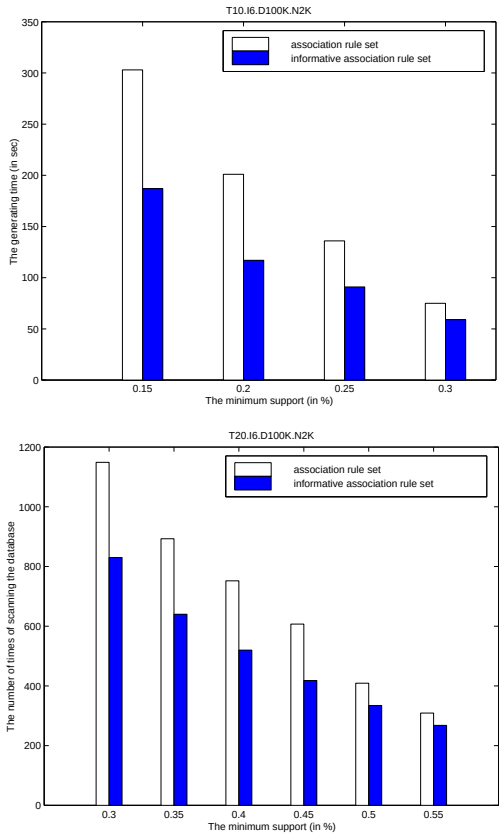

Figure 3. Generating time for different rule sets

The generating time for association rule sets and informative rule sets is listed in the Figure 3. We can see that mining an informative rule set is more efficient than mining a single target association rule set. This is because the informative rule miner 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 excluded from an association rule set as shown in Figure 2.

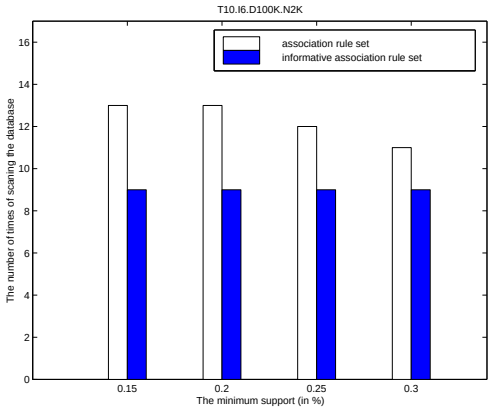

Figure 4. The number of times for scanning the database

Further, the number of times of scanning a database for generating an informative rule set is smaller than that for an association rule set, as showed in Figure 4. This is because the proposed algorithm avoids 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 excluded by an informative rule set than short rules. Clearly, this number is significantly smaller than the number of different antecedents in the generated rule set which are needed to scan a database in aother direct algorithm.

To better understand of improvement of efficiency of the algorithm for mining the informative rule set over that for the association rule set, we list the number of nodes in a candidate tree for both rule sets in Figure 5. They are all frequent itemsets for the association rule set and partial frequent itemsets searched by mining the informative rule set. We can see that in mining the informative rule set, the searched itemsets is less than all frequent itemsets for forming association rules. So, this is the reason for efficiency improvement and reduction in number of scanning a database.

7 Conclusions

We have defined a new, informative, rule set that generates prediction sequences equal to those generated by the association rule set by the confidence priority. The informative rule set is significantly smaller than the association rule set, especially when the minimum support is small. We have studied the upward closed 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.

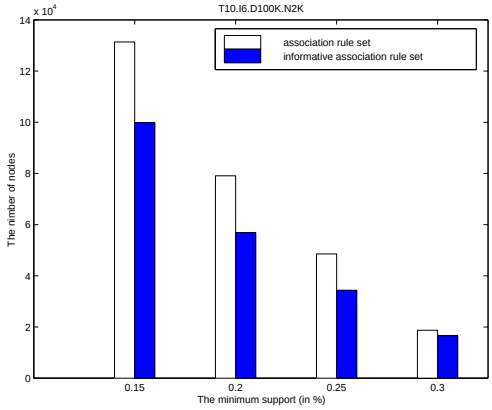

Figure 5. The number of candidate nodes

The experimental results confirm that the informative rule set is significantly smaller than both the association rule set and the non-redundant association rule set, that can be generated more efficiently than the association rule set. The experimental results also show that this efficiency improvement results from that the generation of the informative rule set needs fewer candidates and database accesses than that of the association rule set. The number of database accesses of the proposed algorithm is much smaller than other direct methods for generating unconstrained association rule sets.

Although the informative rule set provides the same prediction sequence as the association rule set, there may exist other definitions of "interesting" in different applications. How to use the informative rule set to make predictions under such different criteria remains a subject of future work.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining associations between sets of items in massive databases. In *Proc. of the ACM SIGMOD Int'l Conference on Management of Data*, 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the Twentieth International Conference on Very Large Databases*, pages 487–499, Santiago, Chile, 1994.

[3] R. Bayardo and R. Agrawal. Mining the most interesting rules. In S. Chaudhuri and D. Madigan, editors, *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 145–154, N.Y., Aug. 15–18 1999. ACM Press.

[4] R. Bayardo, R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense database. In *Proc. of the 15th Int'l Conf. on Data Engineering*, pages 188–197, 1999.

[5] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00)*, pages 1 – 12, May, 2000.

[6] M. Holsheimer, M. Kersten, H. Mannila, and Toivonen. A perspective on databases and data mining. In *1st Intl. Conf. Knowledge Discovery and Data Mining*, Aug. 1995.

[7] M. Houtsma and A. Swami. Set-oriented mining of association rules in relational databases. In *11th Intl. Conf. data Engineering*, 1995.

[8] B. Liu, W. Hsu, and Y. Ma. Pruning and summarizing the discovered associations. In *SIGKDD 99*, 1999.

[9] H. Mannila, H. Toivonen, and I. Verkamo. Efficient algorithms for discovering association rules. In *AAAI Wkshp. Knowledge Discovery in Databases*, July 1994.

[10] R. Ng, L. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained associations rules. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD-98)*, volume 27 of *ACM SIGMOD Record*, pages 13–24, New York, June1–4 1998. ACM Press.

[11] J. S. Park, M. Chen, and P. S. Yu. An effective hash based algorithm for mining association rules. In *ACM SIGMOD Intl. Conf. Management of Data*, May 1995.

[12] R. Rymon. Search through systematic set enumeration. In W. Nebel, Bernhard; Rich, Charles; Swartout, editor, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 539–552, Cambridge, MA, oct 1992. Morgan Kaufmann.

[13] A. Savasere, R. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In *21st VLDB Conf.*, 1995.

[14] P. Shenoy, J. R. Haritsa, S. Sudarshan, G. Bhalotia, M. Bawa, and D. Shah. Turbo-charging vertical mining of large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD-99)*, pages 22–33.

[15] H. Toivonene, M. Klemettinen, P. RonKainen, K. Hatonen, and H. Mannila. Pruning and grouping discovered association rules. Technical report, Department of Computer Science, University of Helsinki, Finland, 1998.

[16] G. I. Webb. Efficient search for association rules. In R. Ramakrishnan, S. Stolfo, R. Bayardo, and I. Parsa, editors, *Proceedinmgs of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-00)*, pages 99–107, N. Y., Aug. 20–23 2000. ACM Press.

[17] M. J. Zaki. Generating non-redundant association rules. In *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 34 – 43, August 2000.

[18] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, page 283. AAAI Press, 1997.