PERGAMON

Computers & Industrial Engineering 43 (2002) 721–733

computers & industrial engineering

www.elsevier.com/locate/dsw

Simple association rules (SAR) and the SAR-based rule discovery

Guoqing Chen[a,*], Qiang Wei[a], De Liu[b], Geert Wets[c]

[a]*School of Economics and Management, Tsinghua University, Beijing 100084, People's Republic of China*
[b]*Center for Research on E-Commerce, University of Texas at Austin, Austin, TX 78712, USA*
[c]*Limburg University, Universitaire Campus Bld D, 3590 Diepenbeek, Belgium*

Abstract

Association rule mining is one of the most important fields in data mining and knowledge discovery in databases. Rules explosion is a problem of concern, as conventional mining algorithms often produce too many rules for decision makers to digest. Instead, this paper concentrates on a smaller set of rules, namely, a set of simple association rules each with its consequent containing only a single attribute. Such a rule set can be used to derive all other association rules, meaning that the original rule set based on conventional algorithms can be 'recovered' from the simple rules without any information loss. The number of simple rules is much less than the number of all rules. Moreover, corresponding algorithms are developed such that certain forms of rules (e.g. 'P ⇒ ?' or '? ⇒ Q') can be generated in a more efficient manner based on simple rules.

Keywords: Data mining; KDD; Simple association rules

1. Introduction

Data mining, as one of the promising technologies since 1990s, is to some extent a non-traditional data-driven method to discover novel, useful, hidden knowledge from massive data sets. It has been considered very important in business, industries and engineering. Data mining can be categorized into several interesting areas, such as clustering, association rules, decision tree analysis, prediction, regression, etc. (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). In particular, since Agrawal, Imielinski, and Swarmi (1993) introduced the notion of association rules in 1993, association rule mining has attracted more and more attention of academia and practitioners with applications such as customer relation management (CRM), market baskets, economic and financial time-series analysis, production process, manufacturing diagnosis, etc. (Fayyad et al., 1996; Motwani, Ullman & Tsur, 1997; Wang, Chen, & Cai, 1997).

* Corresponding author. Tel.: +86-10-6277-2940; fax: +86-10-6278-5876.
E-mail address: chengq@em.tsinghua.edu.cn (G. Chen).

0360-8352/02/$ - see front matter
PII: S0360-8352(02)00135-3

Roughly speaking, an association rule can be regarded as a relationship of the form $A \Rightarrow B$, where A and B are two separate sets of items. Two measures, namely the degree of support (Dsupp) and the degree of confidence (Dconf), are used to define a rule. Dsupp is the percentage of transactions containing both A and B in the whole data set. Dconf is the ratio of the number of transactions that contain A and B over the number of transactions that contain A. For example, a rule like "Milk $\Rightarrow$ Diapers with Dsupp = 20%, Dconf = 80%" means that "20% of the customers bought both Milk and Diapers" and "80% of the customers who bought Milk also bought Diapers".

Thus, the process of mining association rules is, given pre-defined thresholds (i.e. minimal support and minimal confidence), to search the whole data set and discover all possible association rules with their Dsupps and Dconfs greater than or equal to the thresholds. In accordance with this mining process, the basic Apriori algorithm, regarded as a conventional mining method, was introduced by Agrawal and Srikant (1994) and Usama, Fayyad, and Uthurusamy (1994). Many research efforts have then been made in two directions. One is to extend the notion of association rules, giving rise to various extensions such as generalized association rules (Srikant & Agrawal, 1995), quantitative association rules (Srikant & Agrawal, 1996), fuzzy association rules (Chen, Wei, & Kerre, 2000; Kuok, Fu, & Wong, 1998; Wei & Chen, 1999), etc. The other is to improve the algorithms in various ways such as fast algorithms (Agrawal & Srikant, 1994; Zaki, Parthasarathy, Ogihara, & Li, 1999), sampling algorithms (Usama et al., 1994), parallel and distributed algorithms (Agrawal & Shafer, 1996; Mueller, 1995), etc.

Notably, in the above-mentioned Apriori-based mining process, a large number of combinations of items need to be scanned and generated, usually resulting in so many rules that cannot easily be handled or used by decision makers. Therefore, 'rule explosion' itself becomes a problem, which can be even severe in dense data sets (Roberto, Bayardo, Agrawal, & Gunopulos, 1999b). With the resultant rule set (interchangeably, hereafter referred to as the original rule set; otherwise indicated where necessary), the problem could be dealt with in terms of rule interestingness, which is aimed at filtering out those useless, redundant, or conflicting rules from the original rule set.

In doing so, not only are appropriate navigation mechanisms and visualization tools needed, but also new appropriate measures for defining interestingness are required. There have been a variety of attempts on rule interestingness from different perspectives (Fukuda, Morimoto, & Morishita, 1996; Klemettinen, Mannila, Ronkainen, Toivonen, & Verkamo, 1994; Motwani et al., 1997; Roberto, Bayardo, & Agrawal, 1999a; Roberto et al., 1999b; Srikant, Vu, & Agrawal, 1997). For example, the measures such as gain and laplace evaluate rules according to their predictive strength. The measures such as interest, conviction and improvement evaluate rules according to their predictive advantage. For example, if we already have '$A \Rightarrow B$ with Dconf being 86%', then '$AC \Rightarrow B$ with Dconf being 80%' is seemingly uninteresting, since the latter does not provide any significant predictive advantage over the former. Other measures such as template can help filter out the rules that users do not care. Recently, from the viewpoint of rule quality, a worth-noting work has been carried out to efficiently derive certain rules that are of 100% confidence using a partition-based randomized algorithm (Yilmaz, Triantaphyllou, Chen, & Liao, 2002). These interestingness measures and efforts can reduce the number of extracted rules and improve the quality of rules to some extent.

While gain, laplace, interest, conviction, improvement, predictive strength and advantage are defined or specified by users or experts, a template needs more a user's participation to filter out uninteresting rules. On one hand, these measures are user-oriented and context-dependent, which are deemed sound and intuitive; on the other hand, many of them are heuristic and are based upon preset criteria or somewhat subjective assumptions. Furthermore, the interesting rules may cause certain information loss.

Here by information loss we mean that the semantics reflected by the original rule set cannot be represented by the semantics reflected by the obtained interesting rules. This is due to the fact that the original rule set cannot be recovered by the rules obtained from filtering based on interestingness measures or preset criteria.

Furthermore, in some cases, decision makers may be interested in certain forms of rules instead of all the rules, such as 'high demand results in …' ('$P \Rightarrow ?$') or '…cause machine malfunctioning' ('$? \Rightarrow Q$'). Such focused rules can be found from all the rules generated by conventional mining methods. Of course, it will be desirable if the available information (e.g. known P or Q) could be incorporated into the rule generation process so as to derive the rules more effectively.

In accordance with these concerns, the notion of simple association rules (SAR) will be introduced in this paper. A simple rule is the rule with a single item as its consequent. It has been observed that Dconf of a rule with multiple items in its consequent could be represented by Dconfs of other rules each with a single item in its consequent. For instance,

$$\begin{aligned}\text{Dconf}(\text{printer} \Rightarrow \text{paper\&folder}) &= \frac{\|\text{printer\&paper\&folder}\|}{\|\text{printer}\|} \\ &= \frac{\|\text{printer\&paper\&folder}\|}{\|\text{printer\&paper}\|} \times \frac{\|\text{printer\&paper}\|}{\|\text{printer}\|} \\ &= \text{Dconf}(\text{printer\&paper} \Rightarrow \text{folder}) \times \text{Dconf}(\text{printer} \Rightarrow \text{paper})\end{aligned}$$

Hereby $\|X\|$ denotes the number of transactions containing the set of items, X, in the database. Thus, one may first concentrate on mining simple rules, based on which other rules concerned can be derived. Importantly, the set of simple rules is smaller in size but as 'equivalently' rich in semantics as the original rule set. That is, such a set of simple rules can be used to induce all rules without information loss, as well as those specifically focused rules. Working with simple rules has several advantages. First, in some cases, simple rules are enough, e.g. using association rules for classification purposes. Second, generating the focused rules does not need to obtain all rules. Third, deriving all the rules based on simple rules involves a fewer number of candidate rules to be generated and evaluated. Please note that the procedures discussed by Yilmaz et al. (2002) also derive SAR.

This paper is organized as follows. In Section 2, the simple rule set is defined along with its properties. In Section 3, the method for mining simple rules is discussed. Section 4 deals with generating other rules based on simple rules.

2. Problem statement and definitions

2.1. Preliminary description

Mining of association rules was introduced by Agrawal et al. (1993). Let $I = \{I_i, i = 1, \ldots, m\}$ be a set of literals, called items. A database D is a set of transactions, where each transaction t is a set of items such that $t \subseteq I$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \varnothing$. A transaction t is called to contain X, if $X \subseteq t$. Let Dsupp(X) be the fraction of transactions that contain X in a database D. The degree of support for a rule $X \Rightarrow Y$ is defined as Dsupp($X \Rightarrow Y$) =

$\mathrm{Dsupp}(X \cup Y)$. The degree of confidence for $X \Rightarrow Y$ is defined as $\mathrm{Dconf}(X \Rightarrow Y) = \mathrm{Dsupp}(X \cup Y)/\mathrm{Dsupp}(X)$. The problem of mining association rules is to find all association rules that have their degrees of support and of confidence no less than the pre-specified minimal support α and minimal confidence β, respectively. In this paper, we use Ψ to denote the set of all discovered rules, i.e.

$$\Psi = \{r : X \Rightarrow Y | \mathrm{Dsupp}(r) \geq \alpha, \mathrm{Dconf}(r) \geq \beta, X \subset I, Y \subset I, \text{ and } X \cap Y = \varnothing\}.$$

2.2. Simple association rules

Unlike functional dependencies, association rules generally are not transitive and do not compose. For example, given that $X \Rightarrow Y$ and $X \Rightarrow Z$ hold in database D, one cannot conclude that $X \Rightarrow YZ$ also holds in database D. However, the following fact holds for association rules: $\mathrm{Dconf}(X \Rightarrow YZ) = \mathrm{Dconf}(X \Rightarrow Y) \times \mathrm{Dconf}(XY \Rightarrow Z)$, which is stated as Lemma 1.

Lemma 1. *Given $X, Y, Z \subset I$, $X \cap Y = \varnothing$, $Y \cap Z = \varnothing$, $X \cap Z = \varnothing$, we have*

1. $\mathrm{Dconf}(X \Rightarrow YZ) = \mathrm{Dconf}(X \Rightarrow Y) \times \mathrm{Dconf}(XY \Rightarrow Z) = \mathrm{Dconf}(X \Rightarrow Z) \times \mathrm{Dconf}(XZ \Rightarrow Y)$
2. $\mathrm{Dsupp}(X \Rightarrow YZ) = \mathrm{Dsupp}(XY \Rightarrow Z) = \mathrm{Dsupp}(XZ \Rightarrow Y)$

Proof. According to the definition of Dconf, we have

$$\mathrm{Dconf}(X \Rightarrow YZ) = \frac{\mathrm{Dsupp}(XYZ)}{\mathrm{Dsupp}(X)} = \frac{\mathrm{Dsupp}(XY) \times \mathrm{Dconf}(XY \Rightarrow Z)}{\mathrm{Dsupp}(X)}$$

$$= \mathrm{Dconf}(X \Rightarrow Y) \times \mathrm{Dconf}(XY \Rightarrow Z)$$

By exchanging Y and Z, we obtain,

$$\mathrm{Dconf}(X \Rightarrow YZ) = \mathrm{Dconf}(X \Rightarrow Z) \times \mathrm{Dconf}(XZ \Rightarrow Y)$$

According to the definition of Dsupp, we have

$$\mathrm{Dsupp}(X \Rightarrow YZ) = \mathrm{Dsupp}(XYZ) = \mathrm{Dsupp}(XY \Rightarrow Z) = \mathrm{Dsupp}(XZ \Rightarrow Y). \qquad \square$$

Consequently, we immediately have the following:

Lemma 2. *Given X, Y, Z, Ψ, where $X, Y, Z \subset I$, $X \cap Y = \varnothing$, $Y \cap Z = \varnothing$, $X \cap Z = \varnothing$, and Ψ is the set of all discovered rules, then if $X \Rightarrow YZ \in \Psi$, then $X \Rightarrow Y \in \Psi$, $X \Rightarrow Z \in \Psi$, $XY \Rightarrow Z \in \Psi$ and $XZ \Rightarrow Y \in \Psi$.*

Proof. According to Lemma 1, we have $\mathrm{Dconf}(X \Rightarrow YZ) = \mathrm{Dconf}(X \Rightarrow Y) \times \mathrm{Dconf}(XY \Rightarrow Z)$. Since $\mathrm{Dconf}(X \Rightarrow Y) \geq \mathrm{Dconf}(X \Rightarrow YZ)$ and $\mathrm{Dconf}(XY \Rightarrow Z) \geq \mathrm{Dconf}(X \Rightarrow YZ)$, and $\mathrm{Dsupp}(XY \Rightarrow Z) = \mathrm{Dsupp}(X \Rightarrow YZ) \leq \mathrm{Dsupp}(X \Rightarrow Y)$, then from $X \Rightarrow YZ \in \Psi$, we can conclude that $X \Rightarrow Y \in \Psi$ and $XY \Rightarrow Z \in \Psi$. Likewise, $X \Rightarrow Z \in \Psi$ and $XZ \Rightarrow Y \in \Psi$. $\square$

According to Lemma 1, if Dsupps and Dconfs of $X \Rightarrow Y$ and $XY \Rightarrow Z$ are known, then Dsupp and Dconf of $X \Rightarrow YZ$ can be computed directly. In addition, Lemma 2 tells us that if $X \Rightarrow YZ$ is a qualified association rule (i.e. $X \Rightarrow YZ \in \Psi$), then so are $X \Rightarrow Y$ and $XY \Rightarrow Z$. With these results, one may further consider that, if all such qualified rules can be decomposed into simple rules, then it would be desirable to find a subset of rules from which all other rules in Ψ may be derived. Theorem 1 states that all association rules can indeed be 'derived' from a simple rule set. Prior to Theorem 1, the notion of 'derive' is described as follows.

Definition 1. Let R be an association rule set. A rule r is called to be derived from R, denoted by $r \in_{\mathrm{d}} R$, if $r \in R$ or Dsupp and Dconf of rule r can be expressed by Dsupps and Dconfs of rules in R in the form of Lemma 1.

Apparently, all rules in R can be derived from R.

Theorem 1. *For a given database and pre-specified minimal support α and confidence β, let Ψ denote the set of all rules*: $\Psi = \{r : X \Rightarrow Y | X, Y \subset I, X \cap Y = \varnothing, \mathrm{Dsupp}(r) \geq \alpha, \mathrm{Dconf}(r) \geq \beta\}$, *and let* Ψ_{s} *denote the simple rule set*: $\Psi_{\mathrm{s}} = \{r : X \Rightarrow A | X \subset I, A \in I, X \cap A = \varnothing, \mathrm{Dsupp}(r) \geq \alpha, \mathrm{Dconf}(r) \geq \beta\}$, *then*

1. *If* $r \in \Psi$, *then* $r \in_{\mathrm{d}} \Psi_{\mathrm{s}}$,
2. $\{r | r \in_{\mathrm{d}} \Psi_{\mathrm{s}}, \mathrm{Dsupp}(r) \geq \alpha, \mathrm{Dconf}(r) \geq \beta\} = \Psi$.

Proof. (1) First consider a rule r with one item in its consequent: $X \Rightarrow I_i$ in Ψ. Obviously r is also a member of Ψ_{s}. Now consider a rule r in Ψ with k items in its consequent, without loss of generality: $X \Rightarrow I_1 I_2 \cdots I_k$, $k \geq 2$. According to Lemma 1, $\mathrm{Dconf}(r) = \mathrm{Dconf}(X \Rightarrow I_1)\mathrm{Dconf}(XI_1 \Rightarrow I_2 I_3 \cdots I_k)$. According to Lemma 2 and $X \Rightarrow I_1 I_2 \cdots I_k \in \Psi$, we have $X \Rightarrow I_1, XI_1 \Rightarrow I_2 I_3 \cdots I_k \in \Psi$. Then by applying Lemma 1 to $XI_1 \Rightarrow I_2 I_3 \cdots I_k$, $\mathrm{Dconf}(r) = \mathrm{Dconf}(X \Rightarrow I_1) \times \mathrm{Dconf}(XI_1 \Rightarrow I_2) \times \mathrm{Dconf}(XI_1 I_2 \Rightarrow I_3 \cdots I_k)$. Again, applying Lemma 2 to $XI_1 \Rightarrow I_2 I_3 \cdots I_k$ will lead to $XI_1 \Rightarrow I_2 \in \Psi$, $XI_1 I_2 \Rightarrow I_3 \cdots I_k \in \Psi$. Similarly, repeatedly using Lemma 1 and Lemma 2, we will finally get

$$\mathrm{Dconf}(r) = \mathrm{Dconf}(X \Rightarrow I_1) \times \mathrm{Dconf}(XI_1 \Rightarrow I_2) \times \cdots \times \mathrm{Dconf}(XI_1 I_2 \cdots I_{k-1} \Rightarrow I_k)$$

and $X \Rightarrow I_1, XI_1 \Rightarrow I_2, \ldots, XI_1 I_2 \cdots I_{k-1} \Rightarrow I_k \in \Psi$. Since $X \Rightarrow I_1, XI_1 \Rightarrow I_2, \ldots, XI_1 I_2 \cdots I_{k-1} \Rightarrow I_k$ are members of Ψ_{s}, we can conclude that the Dconf of r can be computed by Dconfs of rules in Ψ_{s}.

Now examine the Dsupp of r. Since $\mathrm{Dsupp}(r) = \mathrm{Dsupp}(XI_1 I_2 \cdots I_{k-1} I_k) = \mathrm{Dsupp}(XI_1 I_2 \cdots I_{k-1} \Rightarrow I_k)$, it means that Dsupp of r equals Dsupp of $XI_1 I_2 \cdots I_{k-1} \Rightarrow I_k$ which is a member of Ψ_{s}.
Since both Dsupp and Dconf of r can be expressed by Dsupps and Dconfs of rules in Ψ_{s}, we conclude that $r \in_{\mathrm{d}} \Psi_{\mathrm{s}}$.
(2) On one hand, if $r \in \Psi$ then $\mathrm{Dsupp}(r) \geq \alpha$ and $\mathrm{Dconf}(r) \geq \beta$ by definition. Also according to (1), we have $r \in_{\mathrm{d}} \Psi_{\mathrm{s}}$. That is, $r \in \Psi \Rightarrow r \in_{\mathrm{d}} \Psi_s, \mathrm{Dsupp}(r) \geq \alpha, \mathrm{Dconf}(r) \geq \beta$. On the other hand, if $r \in_{\mathrm{d}} \Psi_{\mathrm{s}}$, and $\mathrm{Dsupp}(r) \geq \alpha, \mathrm{Dconf}(r) \geq \beta$ then it is straightforward that $r \in \Psi$. That is, $r \in_{\mathrm{d}} \Psi_{\mathrm{s}}, \mathrm{Dsupp}(r) \geq \alpha, \mathrm{Dconf}(r) \geq \beta \Rightarrow r \in \Psi$. Thus, $\{r | r \in_{\mathrm{d}} \Psi_{\mathrm{s}}, \mathrm{Dsupp}(r) \geq \alpha, \mathrm{Dconf}(r) \geq \beta\} = \Psi$. □

Theorem 1 states that (i) all rules in the whole rule set can be derived from the simple rule set; (ii)

Table 1
An example of a data set

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

applying minimal support and confidence thresholds onto the derived rules will result in exactly the whole rule set.

2.3. An example

The following example is presented to help illustrate the previous ideas.

From a given data set (Table 1) totally 18 rules will be discovered traditionally, with $\alpha = 3/7$, $\beta = 65\%$ (Table 2). Among these rules, #1–#14 are simple rules for each of them has a single item in its consequent. Now look at composite consequent rules #15–#18, they can all be derived from single consequent rules (Table 3). Then, in the process of mining association rules, one does not need to scan the data set to compute Dsupps and Dconfs of #15, #16, #17 and #18, but only needs to compute their corresponding simple rules and finally derive the composite consequent rules using these simple rules,

Table 2
All discovered rules ($\alpha = 3/7$, $\beta = 65\%$)

Rules	Expression	Dsupp	Dconf (%)
#1	$A \Rightarrow B$	3/7	75
#2	$A \Rightarrow D$	3/7	75
#3	$A \Rightarrow E$	3/7	75
#4	$B \Rightarrow D$	4/7	66.7
#5	$D \Rightarrow B$	4/7	80
#6	$B \Rightarrow E$	6/7	100
#7	$E \Rightarrow B$	6/7	100
#8	$D \Rightarrow E$	4/7	80
#9	$E \Rightarrow D$	4/7	66.7
#10	$AB \Rightarrow E$	3/7	100
#11	$AE \Rightarrow B$	3/7	100
#12	$BD \Rightarrow E$	4/7	100
#13	$BE \Rightarrow D$	4/7	66.7
#14	$DE \Rightarrow B$	4/7	100
#15	$A \Rightarrow BE$	3/7	75
#16	$B \Rightarrow DE$	4/7	66.7
#17	$D \Rightarrow BE$	4/7	80
#18	$E \Rightarrow BD$	4/7	66.7

Table 3
Derived rules

Rules	Derived from
#15: $A \Rightarrow BE$	{#1: $A \Rightarrow B$, #10: $AB \Rightarrow E$}
	{#3: $A \Rightarrow E$, #11: $AE \Rightarrow B$}
#16: $B \Rightarrow DE$	{#4: $B \Rightarrow D$, #12: $BD \Rightarrow E$}
	{#6: $B \Rightarrow E$, #13: $BE \Rightarrow D$}
#17: $D \Rightarrow BE$	{#5: $D \Rightarrow B$, #12: $BD \Rightarrow E$}
	{#8: $D \Rightarrow E$, #14: $DE \Rightarrow B$}
#18: $E \Rightarrow BD$	{#7: $E \Rightarrow B$, #13: $BE \Rightarrow D$}
	{#9: $E \Rightarrow D$, #14: $DE \Rightarrow B$}

which are illustrated in Table 3. That is, mining the simple rule set instead of the whole rule set will be more efficient and easier.

3. Mining simple association rules

Based on the notion of simple rules, this section focuses on the corresponding mining algorithm.

3.1. Basic ideas

The process could be decomposed into two phases (Agrawal et al., 1993; Agrawal & Srikant, 1994):

1. Find all frequent itemsets, i.e. the itemsets that satisfy the minimal support threshold. This phase is the basic step and similar to Apriori algorithm.
2. For each frequent itemset $X = I_1 I_2 \cdots I_k$, compute Dconf(r) for each potential rule $r : X - I_j \Rightarrow I_j$, $j = 1, \ldots, k$. If Dconf(r) is greater than or equal to minimal confidence, then output r.

The difference between mining for SAR and mining for all association rules is that, to extract SAR from any k-frequent itemset $\{I_1, I_2, \ldots, I_k\}$, i.e. the frequent itemset consisting of k items, one only needs to examine certain frequent itemsets, namely its $(k-1)$-frequent sub-itemsets instead of all j-frequent sub-itemsets ($j = 1, 2, \ldots, k-1$). This leads to a substantial reduction of the computations. More concretely, for a certain k-frequent itemset, traditionally, one needs to check all its (k − 1)-frequent sub-itemsets, (k − 2)-frequent sub-itemsets, …, and l-frequent itemsets. That is, totally $(2^k - 2)$ sub-itemsets should be checked. Thus, for each k-frequent itemset, the Apriori algorithm computes Dconfs for $(2^k - 2)$ potential rules that are comprised of exactly k items. In the case of the simple rule mining, however, for each k-frequent itemset, only $k(k-1)$-frequent sub-itemsets need to be checked. Accordingly, the proposed simple rule mining algorithm computes Dconfs only for k potential rules, instead of $(2^k - 2)$ rules.

3.2. The algorithm

As mentioned in Section 3.1, the mining algorithm only needs to consider $(k-1)$-frequent

```
// The algorithm is to derive the simple rule set based on frequent itemsets.
// Given the minimal confidence threshold: min-confidence
// F_k is the set of k-frequent itemsets
// f^k is a k-frequent itemset of F_k
1   for each f^k ∈ F_k, k ≥ 2;
2       SB = {(k-1)-itemsets f^(k-1) | f^(k-1) ⊂ f^k};
3       for each f^(k-1) ∈ SB
4           Dconf = Dsupp(f^k) / Dsupp(f^(k-1));
5           if Dconf ≥ min-confidence then output r: f^(k-1)⇒(f^k – f^(k-1)) with
            Dconf, Dsupp;
6       endfor;
7   endfor;
```

Fig. 1. Algorithm Rule_Generation.

sub-itemsets for each k-frequent itemset in extracting l-consequent rules. This significantly reduces the number of frequent itemsets needed in the rule generation. Furthermore, the recursive operations, which are used in the Apriori algorithm to generate all rules, will be no longer needed (Agrawal & Srikant, 1994). Specifically, the algorithm is shown as in Fig. 1.

Then with the algorithm Rule_Generation given in Fig. 1, all the simple rules can be derived. Theoretically, for a k-frequent itemset, using the traditional Apriori method (Agrawal & Srikant, 1994), under the best situation (in that all of its corresponding candidate rules whose antecedents are $(k-1)$-subsets and whose consequents are singletons do not satisfy minimal confidence, and therefore the consecutive recursive processes will be no longer necessary), the number of rules that need to be considered is k and only k $(k-1)$-subsets will be loaded into memory. While under the worst situation (in that all of the corresponding rules generated by this k-frequent itemset satisfy the minimal confidence threshold), the number of rules that need to be considered is $2^k - 2$ and correspondingly, all the non-empty and strict $(2^k - 2)$ subsets of this k-frequent itemset need to be loaded into memory. However, in the SAR-based method, for any k-frequent itemset, the number of rules that need to be considered is k and only k $(k-1)$-subsets need to be loaded into memory. Clearly, mining based on SAR is advantageous in efficiency.

To illustrate the effect of reduction in computation, an experiment has been conducted on the

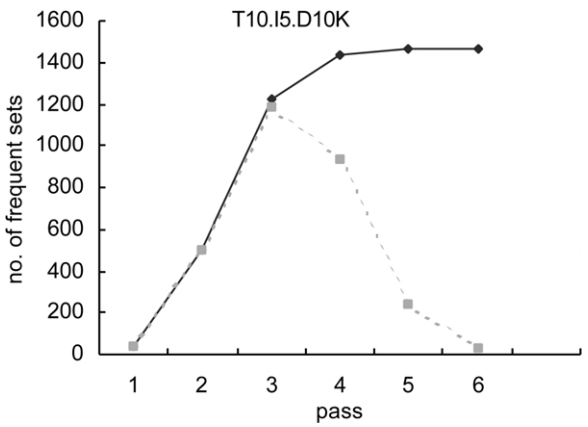

Fig. 2. Numbers of frequent itemsets needed in rule generation.

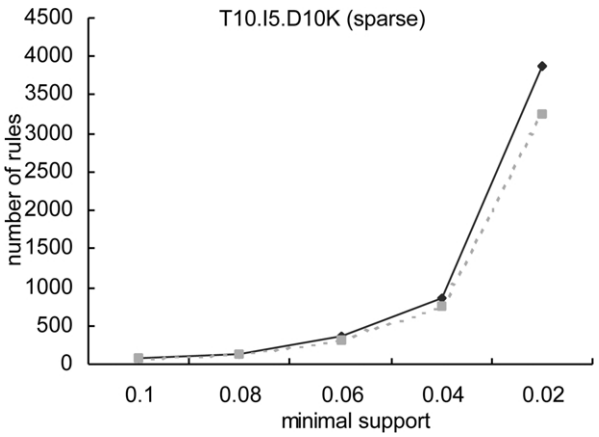

Fig. 3. Experiment on sparse data set.

T10.I5.D10k synthetic data set, which was generated according to a typical procedure as introduced by Agrawal and Srikant (1994). Conventionally, T10.I5.D10k denotes a data set with 10 different items and 10,000 transactions, each being of the length of five items on average. Fig. 2 shows the experimental results. In the case of mining the simple rule set, the number of frequent itemsets (marked by the dashed line) dropped sharply after pass 3, whereas in the case of mining the whole rule set, the number of frequent itemsets (marked by the solid line) increased continuously as the number of passes increases.

3.3. The number of rules

Since the set of simple rules is a subset of the set of all rules, the number of simple rules is usually smaller than that of all rules. Theoretically, in the best case, i.e. when all possible rules pertain, which means that the thresholds minimal support $= 0$, minimal confidence $= 0$, there will be a total of $(3^m - 2^{m+1} + 1)$ rules generated. By contrast, there will be only $m(2^{m-1} - 1)$ simple rules generated. Hereby, $m = |I|$ is the number of items. For instance, with 10 items, there will be a total of 57,002 rules generated, whereas there will be only 5,110 simple rules. In the worst case, the set of all the rules generated equals the set of simple rules.

Further, two synthetic data sets were used to compare the number of rules. To simulate the data sets with different density (based on Agrawal & Srikant, 1994), the number of potential frequent itemsets was set to different levels. In these experiments, 50 for dense data and 100 for sparse data were used. To examine the effect of different minimal support levels, the minimal confidence is fixed at 60%.

The experiment results (i.e. Figs. 3 and 4) revealed that the number of simple rules was usually 10–50% smaller than that of all rules. This gap would increase as the minimal support threshold decreased. On dense data sets there would be a relatively small set of simple rules discovered.

4. Deriving focused rules based on simple rules

In some cases, e.g. when using association rules for classification purposes, simple rules are sufficient. In others cases, however, one may need to know other rules such as '$P \Rightarrow ?$' and '$? \Rightarrow Q$'. Instead of

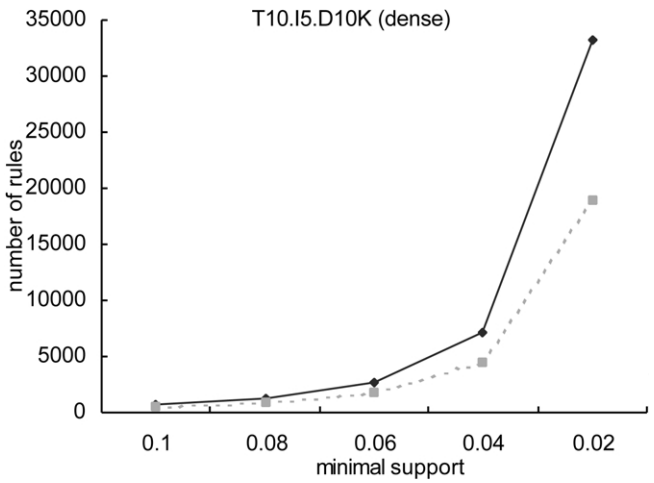

Fig. 4. Experiment on the dense data set.

discovering such rules by generating and then screening all rules according to conventional mining methods, one could derive the rules using the given antecedent or consequent based on simple rules.

Consider the case of '$P \Rightarrow ?$'. For example, given the antecedent {sex = female and education = high}, one needs to find all rules with antecedent {sex = female}, {education = high} or {sex = female and education = high}. With the set of simple rules Ψ_s and an antecedent $P(P \subseteq I)$, Algorithm 4.1 will derive all rules $\{r : X \Rightarrow Y\}$, such that $X \subseteq P$ (Fig. 5).

For each k-consequent rule r in Π_k, the algorithm will scan Ψ_s for rules whose antecedents are exactly the combination of the antecedent and consequent of r. If they also satisfy the minimal support and minimal confidence thresholds, then they are used to derive new $(k+1)$-consequent rules. Concretely, if we derive Π_k, then for each $X \Rightarrow I_1 I_2 \cdots I_k, I_j \in I, j = 1, 2, \ldots, k$, we scan Ψ_s to see if there exists any $XI_1 I_2 \cdots I_k \Rightarrow I_{k+1}$, based on which a $(k+1)$-consequent rule could be generated and put into Π_{k+1} if

```
// this algorithm is to derive all rules satisfying antecedent P
// Π_k denotes the set of k-consequent rules
1   Π_1 ← ∅;
2   for each rule r: X⇒I_i in Ψs
3       if X ⊆ P   then Π_1 ← {r, Dsupp (r), Dconf (r)};
4   endfor;
5   k ←1;
6   while Π_k ≠ ∅ do begin
7       Π_{k+1} ← ∅;
8       for each rule r: X⇒I_i1 I_i2 … I_ik, r ∈ Π_k
9           for each rule λ: XI_i1 I_i2 … I_ik ⇒I_ik+1 in Ψ_s
10              if i_{k+1} > i_k and Dconf (r)×Dconf (λ) ≥ β then
11              Π_{k+1} ← {X⇒I_i1 I_i2 … I_ik I_ik+1, Dsupp(λ), Dconf (r)×Dconf (λ)};
12          end for;
13      end for;
14      k←k+1;
15  end while;
```

Fig. 5. Algorithm 4.1.

```
// this algorithm is to derive all rules satisfying consequent Q
// Π_k denotes the set of k-consequent rules
1   Π_1 ← ∅;
2   for each rule r: X⇒I_i in Ψ_s
3       if I_i ∈ Q then Π_1 ← {r, Dsupp (r), Dconf (r)};
4   endfor
5   k ← 1;
6   while Π_k ≠ ∅ do begin
7       Π_{k+1} ← ∅;
8       for each rule r: X⇒I_i1 I_i2 … I_ik in Π_k
9           for each rule λ: XI_i1 I_i2 … I_ik ⇒ I_ik+1 in Π_1
10              if i_{k+1} > i_k and Dconf (r)×Dconf (λ) ≥ β then
11              Π_{k+1} ← {X⇒I_i1 I_i2 … I_ik I_ik+1, Dsupp (λ), Dconf (r) ×Dconf (λ)};
12          end for;
13      end for;
14      k ← k+1;
15  end while;
```

Fig. 6. Algorithm 4.2.

satisfying the minimal confidence threshold. After computing all the k-consequent rules in Π_k, Π_{k+1} is generated. If there exists any element in Π_{k+1}, then continue to generate Π_{k+2}, otherwise stop. In this way, all association rules whose antecedents are subsets of P are derived. Note that an order $i_{k+1} > i_k$ is imposed so as to avoid duplication in the process of derivation.

Consider the case of '$? \Rightarrow Q$'. For example, given the consequent {sex = female and education = high}, one needs to find all rules with consequent {sex = female} or {education = high} or {sex = female and education = high}. With consequent Q, $Q \subseteq I$, the mining algorithm has been developed as shown in Fig. 6.

In the SAR-based mining as shown in algorithms 4.1 and 4.2, the focused rules like '$P \Rightarrow ?$' or '$? \Rightarrow Q$' can be generated. For the sake of simplicity, without loss of generality, consider algorithm 4.1 for the number of rules generated. If $\alpha = 0$, steps 6–13 will lead to a range between $m(2^{m-1} - 1)$ (where only simple rules are qualified rules, so $\Pi_2 = \varnothing$) to $(2^{m-|P|} \times 3^{|P|} - 2^m - 2^{|P|} + 1)$ ($\beta = 0$) depending on given P for $1 \leq |P| \leq m$. Clearly, this is much better than the way to generate all the rules using the conventional methods, which in general are at the level of $(3^m - 2^{m+1} + 1)$. When setting $|P| = |I| = m$, $(2^{m-|P|} \times 3^{|P|} - 2^m - 2^{|P|} + 1) = (3^m - 2^{m+1} + 1)$.

Note that by setting $P = I$ in algorithm 4.1 or $Q = I$ in algorithm 4.2 and evaluating the rules against α and β will result in deriving the whole rule set Ψ. If it is required to generate all the rules in Ψ, either of the algorithms will perform better than the conventional methods. This is true because the number of (candidate) rules generated by any one of algorithms 4.1 and 4.2 will be less than that by the conventional methods. For instance, if some items do not appear in any rule as its consequent, then no other rule with these items will be generated as part of its consequent. This may not be the case for the conventional methods. This could be further illustrated by the following two facts: (1) Dconf(X $\Rightarrow$ Y) $\geq$ Dconf(X $\Rightarrow$ YZ); and (2) Dconf(XY $\Rightarrow$ Z) $\geq$ Dconf(X $\Rightarrow$ YZ). According to Lemma 1, $Dconf(X \Rightarrow YZ) = Dconf(X \Rightarrow Y) \times Dconf(XY \Rightarrow Z)$. Since Dconf(X $\Rightarrow$ Y) and Dconf(XY $\Rightarrow$ Z) $\leq$ 1. So Dconf(X $\Rightarrow$ Y) $\geq$ Dconf(X $\Rightarrow$ YZ) and Dconf(XY $\Rightarrow$ Z) $\geq$ Dconf(X $\Rightarrow$ YZ). In the traditional Apriori method (Agrawal & Srikant, 1994), only the second fact is used as a pruning strategy in the mining process. For any frequent itemset XYI_1, if XY $\Rightarrow$ I_1 does not satisfy the minimal confidence β,

then $X \Rightarrow YI_1$ will not be considered in the process since it does not satisfy β. However, the traditional Apriori method does not incorporate the first fact in the rule-generation algorithm. In the case where $XY \Rightarrow \mathrm{I}_1$ satisfies β but $X \Rightarrow Y$ does not, the traditional Apriori method may still generate candidate rule $X \Rightarrow YI_1$ and test it against β. By contrast, in the SAR-based algorithms 4.1 and 4.2, this situation could be avoided, for if either $X \Rightarrow Y$ or $XY \Rightarrow I_1$ dissatisfies β, then $X \Rightarrow YI_1$ will not be generated as a candidate rule and needs not to be further considered in the mining process.

Consider the example shown in Section 2.3. It is easy to find out that A does not appear as a consequent of any simple rules due to its Dconf value. Therefore, rules containing A on the right hand side (consequent) such as $B \Rightarrow AE$ ($\mathrm{Dsupp}(B \Rightarrow AE) = 3/7$, $\mathrm{Dconf}(B \Rightarrow AE) = 50\%$) and $E \Rightarrow AB$ ($\mathrm{Dsupp}(E \Rightarrow AB) = 3/7$, $\mathrm{Dconf}(E \Rightarrow AB) = 50\%$) will not be generated as candidate rules by algorithm 4.1 or 4.2. However, according to the traditional Apriori method (Agrawal et al., 1994), such rules may be generated as candidate rules because both $AB \Rightarrow E$ and $AE \Rightarrow B$ are qualified rules (though they may be filtered out afterwards according to β). Clearly, the proposed algorithms are more efficient in this regard.

Finally, it is worth noting that the SAR-based method does not involve any heuristics or preset criteria as used in other interestingness measures and related methods. Therefore, information loss could be avoided. That is, the rules obtained based on the SAR method could be used to derive any other rules of the original rule set, if desired so. In addition, if the heuristics are preferred, those interestingness measures may then be applied to the SAR-derived rules, depending on the need of decision makers. However, as indicated previously, those measures may cause information loss.

5. Concluding remarks

Data mining has been widely used in business, industries and engineering. In this paper, a simple rule set has been introduced, which retains all information in the original rule set, but has a smaller size. It has been proved that all qualified association rules can be derived from the SAR. Furthermore, mining algorithms have been proposed to the discovered simple rules, focused rules and all qualified rules. These algorithms have showed their advantages over conventional methods in terms of the number of candidate rules (therefore the time of computation). Currently, ongoing research is being conducted on further algorithmic optimizations. For example, certain pruning strategies are evaluated and will be incorporated into the algorithms.

Acknowledgments

Partly supported by ‘Nation’s Outstanding Young Scientists Funds’ of China (No. 79925001), the Bilateral Scientific and Technological Cooperation Programme Between China and Flandres (174B0201) and Tsinghua’s Soft Science Key Project on E-Commerce.

References

Agrawal, R., Imielinski, T., & Swarmi, A. (1993). Mining association rules between sets of items in large databases. *Proceedings of the ACM-SIGMOD 1993 International Conference on Management of Data, Washington, DC*, 207–216.

Agrawal, R., & Shafer, J. C. (1996). Parallel mining of association rules: design, implementation and experience. *Computer Science/Mathematics*, RJ 10004.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile*, Expanded version available as IBM Research Report RJ9839.

Chen, G. Q., Wei, Q., & Kerre, E. E. (2000). Fuzzy data mining: discovery of fuzzy generalized association rules. *Recent research issues on management of fuzziness in databases. Studies in fuzziness and soft computing*, New York, NY, USA: Springer.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). *From data mining to knowledge discovery: an overview. Advances in knowledge discovery and data mining*, Cambridge, MA, USA: AAAI Press/The MIT Press, pp. 1–30.

Fukuda, T., Morimoto, Y., & Morishita, S. (1996). Data mining using two-dimensional optimized association rules: scheme, algorithms, and visualization. *Proceedings of the 1996 ACM-SIGMOD International Conference on the Management of Data*, 12–13.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., & Verkamo, A. I. (1994). Finding interesting rules from large sets of discovered association rules. *Proceedings of the Third International Conference on Information and Knowledge Management*.

Kuok, C. M., Fu, A., & Wong, M. H. (1998). Mining fuzzy association rules in databases. *SIGMOD Record*, *27*(1), 41–46.

Motwani, B. S., Ullman, J., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. *Proceedings of the 1997 ACM-SIGMOD International Conference on the Management of Data*, 255–264.

Mueller, A. (1995). Fast sequential and parallel algorithms for association rule mining: a comparison. *CS-TR-3515*.

Roberto, J., Bayardo, Jr., & Agrawal, R. (1999a). Mining the most interesting rules. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 145–154.

Roberto, J., Bayardo, Jr., Agrawal, R., & Gunopulos, D. (1999b). Constraint-based rule mining in large dense databases. *Proceedings of the 15th International Conference on Data Engineering*, 188–197.

Srikant, R., & Agrawal, R. (1995). Mining generalized association rules. *Proceedings of the 21st International Conference on Very Large Databases, Zurich, Switzerland*.

Srikant, R., & Agrawal, R. (1996). Mining quantitative association rules in large relational tables. *SIGMOD'96 6/96 Montreal, Canada*.

Srikant, R., Vu, Q., & Agrawal, R. (1997). Mining association rules with item constraints. *Proceedings of the Third International Conference on Knowledge Discovery in Databases and Data Mining, Newport Beach, CA, USA*.

Usama, M., Fayyad, U., & Uthurusamy, R. (1994). Efficient algorithms for discovering association rules. *AAAI Workshop on Knowledge Discovery in Databases, Seattle, Washington, DC, USA*, 181–192.

Wang, Q. Y., Chen, E. H., & Cai, Q. S. (1997). Knowledge discovery and its applications. *Computer Science*, *24*(5).

Wei, Q., & Chen, G. Q. (1999). Mining generalized association rules with fuzzy taxonomic structures. *18th International Conference of NAFIPS, New York, NY, USA*, 477–481.

Yilmaz, E., Triantaphyllou, E., Chen, J., & Liao, T. W. (2002). A heuristic for mining association rules in polynomial time. *Mathematical and Computer Modelling*, *Spring*.

Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1999). New algorithms for fast discovery of association rules. *American Association for Artificial Intelligence*.