算法一：大数据划分挖掘的注释

在“大数据”这个名词出现前，通常使用“大规模数据”和“海量”等来形容数据量的超级大。为了统一名称，本组论文的所有中文注释的描述采用了现在大家熟知的“大数据”这个词，以下不再专门声明。

最初做数据挖掘研究时碰到的第一个难点当然就是数据量大，通常需要几天的时间才能拿到运行结果。所以，我的第一反应就是分而治之！于是，解决这个问题的关键就转变成：如何将划分挖掘的模式融合成能逼近直接挖掘的模式。通过对划分挖掘的模式的仔细分析和理解，我们发现了一种加权融合方法可以逼近直接挖掘的关联模式。它的核心点就是权重的确定：一个关联模式在越多的子集合被挖掘出来，这个关联模式的权值应该越大；一个子集合被挖掘出来的模式中含有权值大的关联模式越多，这个子集合的权值应该越大。这个权重确定的方法得益于美国总统选举办法，每个州的总统选举票数量决定了州的重要性。

请注意，上面提到的“对划分挖掘的模式的理解”是非常重要的，在传统数据挖掘中基本上没有对数据或者数据集合经过理解后再挖掘的算法，通常都是一些应用驱动的挖掘算法。这种理解在某种程度上是对数据集合的理解，导出了一个值得深入探讨的结论：只有对大规模数据划分挖掘并剖析后，才能发现一些新的有用模式，揭示出这些模式的存在形态与内涵。图 1 同时展示了大规模数据的直接挖掘和划分挖掘的过程，后者可以逼近前者的挖掘结果。据我们所知，文献[1]从数学角度证明了这个结论。图 2 展示了划分后在各个数据子集合中发现的局部模式：A 类（蓝色的）、B 类（红色的）和 C 类（绿色的）模式。A 类模式可以采用传统挖掘方法来获取，但是，B 和 C 两类模式是传统算法不能发现的。也就是说，通过这种剖析，我们获得了这样一些新的有趣信息： B 类模式可以用于发现历史数据集合中那些曾经辉煌过的模式，对考证和考古有帮助；C 类模式可以用于发现趋势模式，对市场分析有用。这种理解鼓励了我们展开多源数据（第二篇论文）和动态数据（第三篇论文）挖掘的研究，简单描述如下。

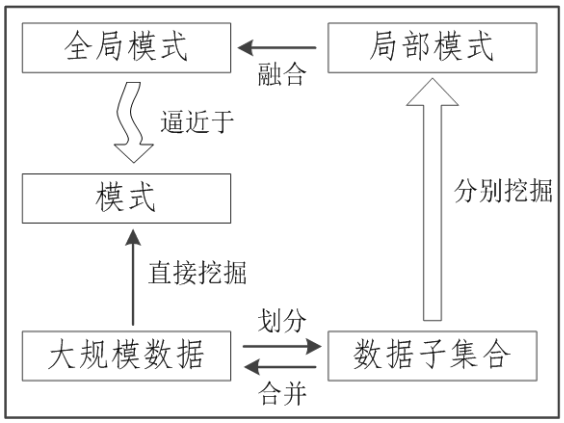

图1 直接挖掘与划分挖掘

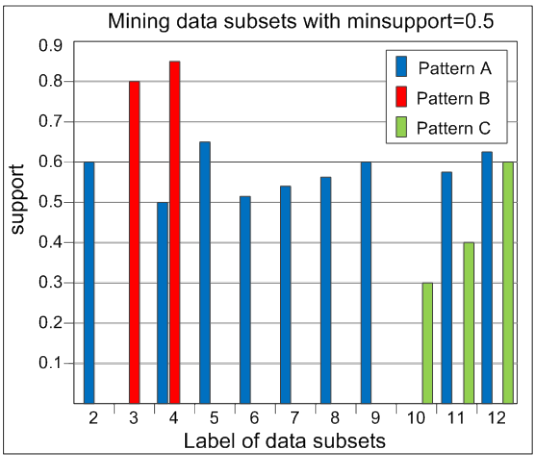

图2 子集合中的模式分布

（1）在多源数据环境下，A、B 和 C 类模式来自于分别挖掘各个数据源，称为

局部模式，它们在局部决策和全局决策中都有重要的应用价值。然而，将数据源集成挖掘的传统方法通常只能发现 A 类型模式中的一部分，因为 B 和 C 两类模式在数据集成后无法识别出来。所以，我们在第二篇论文中提出了局部模式分析（Local Pattern Analysis，LPA）方法，通过基于 LPA 的加权挖掘方法来发现 A、B 和 C 三类模式。

（2）对于一个动态数据集合，我们在第三篇论文中提出了加权挖掘的方法来发现近似的 C 类模式。

参考文献

[1]. C. Xu, Y. Zhang, R. Li, et al. On the Feasibility of Distributed Kernel Regression for Big Data. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(11): 3041-3052.

Applied Artificial Intelligence, 15:129–139, 2001

0883-9514/01 $12.00 + .00

LARGE SCALE DATA MINING BASED ON DATA PARTITIONING

SHICHAO ZHANG
School of Mathematical and Computing Sciences, Guangxi Normal University, Guilin, P.R. China

XINDONG WU
Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado

Dealing with very large databases is one of the defining challenges in data mining research and development. Some databases are simply too large (e.g., with terabytes of data) to be processed at one time. For efficiency and space reasons, partitioning them into subsets for processing is necessary. However, since the number of itemsets in each partitioned data subset can be a combinatorial amount and each of them may be a large itemset in the original database, data mining results from these subsets can be very large in size. Therefore, the key to data partitioning is how to aggregate the results from these subsets. It is not realistic to keep all results from each subset, because the rules from one subset need to be verified for usefulness in other subsets. This article presents a model of aggregating association rules from different data subsets by weighting. In particular, the aggregation efficiency is enhanced by rule selection.

Association analysis in large databases has received much attention recently (Agrawal et al., 1993; Brin, et al., 1997; Srikant & Agrawal, 1997). Let $I = \{i_1, i_2, \ldots, i_N\}$ be a set of N distinct literals called items, and D a set of transactions over I. Each transaction contains a set of items $i_1, i_2, \ldots, i_k \in I$. An association rule is an implication of the form $A \rightarrow B$, where $A, B \subset I$, and $A \cap B = \varnothing$. Each itemset (such as A and B) has an associated statistical measure called support, denoted as supp. For an itemset $A \subset I$, supp(A) = s, if the fraction of transactions in D containing A equals to s. A rule $A \rightarrow B$ has a measure of strength called confidence which is defined as the ratio $\text{supp}(A \cup B)/\text{supp}(A)$. The problem of mining association rules is to generate all rules $A \rightarrow B$ that have both support and confidence greater than or equal to some user specified thresholds, called minimum support and minimum confidence, respectively.

Address correspondence to Xindong Wu, Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO 80401. E-mail: xwu@mines.edu

To implement association analysis, a wide range of problems have been investigated over such diverse topics as models for discovering generalized associated rules (Srikant & Agrawal, 1997), efficient algorithms for computing the support and confidence of an association rule (Park et al., 1995), measurements of interestingness (Agrawal et al., 1993; Brin et al., 1997), mining negative association rules (Brin et al., 1997), and computing large itemsets online (Hidber, 1999). The main limitation of these approaches, however, is that they require multiple passes over the database. For a very large database that is typically disk resident, this requires reading the database completely for each pass resulting in a large number of disk I/Os. Consequently, the larger the size of a given database, the greater the number of disk I/Os. This means that existing models cannot work well when resources are bounded. Therefore, faster mining models have to be explored.

Recently, some sampling models of mining approximate association rules by Chernoff bounds have been proposed in (Srikant & Agrawal, 1997; Toivonen, 1996). As the sample size is typically much smaller than the original database size, the association rules on the sample can be obtained at a much faster time. For example, for a given very large database D with over 10^6 transactions, if one chooses a random subset RD of D as the operating object of mining association rules, which has several thousand transactions, the running time can be minimized. Such a random subset RD of D would maintain that the support of an itemset in RD is approximately equal to that in D. However, sampling models assume that the transactions of a given database are randomly appended to the database in order to hold binomial distribution. Our main motivation in this article is to propose a new model to deal with very large transaction databases. In this model, a given database is first partitioned into several subsets according to allowed resources or requirements. Second, each subset is mined for association rules. Third and most important, we aggregate rules from different data subsets by weighting. Finally, we select high rank rules as the output.

In many fields such as probability and fuzzy set theory, weighting is taken as a main method for aggregating information. For example, consider a diagnosis in a hospital. Let A be a patient, d_1, d_2, d_3, d_4, d_5 be five medical experts in the hospital with weights w_1, w_2, w_3, w_4, w_5, respectively. After diagnosing, the patient is judged to be with one of the following four diseases: s_1, s_2, s_3, and s_4. To determine a final conclusion, it needs to synthesize the diagnosis by these experts. Assume b_{ij} be the belief of the patient with jth disease given by expert d_i, $i = 1, 2, 3, 4, 5$, $j = 1, 2, 3, 4$. Then the belief of the patient with disease s_j is synthesized as p_j where, $j = 1, 2, 3, 4$. According to the synthesis, one can rank diseases s_1, s_2, s_3, s_4 by p_1, p_2, p_3, p_4. The disease with the highest rank is taken as the final result.

To aggregate association rules from data subsets, the above weighting

process is adopted in this paper. In the above diagnosis, the key is to allocate a reasonable weight to each expert. Consequently, we will research how to determine proper weights for our tasks in this article.

To allocate weights, Good (1950) defines the weight in favor of a hypothesis, H, provided by evidence, E: the weight of evidence is calculated in terms of the ratio of the likelihoods. In this definition, he defines a concept "almost as important as that of probability itself." It is obvious that the weight of evidence can be used to evaluate the weights of the discovered rules from the subsets of a given database. Good's idea is applied in this article in such a way that the weight of each rule is almost as important as that of its frequency in the subsets in the model.

To aggregate association rules from the subsets of a given database, one also needs to determine the weight for each subset. According to Good's definition on weight, the more the number of subsets that support the same rule, the larger the frequency of the rule should be (or the larger the weight of the rule should be). In the meantime, if a subset supports a larger number of high-frequency rules, the weight of the subset should also be higher. The goal in this article is to extract the high-frequency rules in the partitioned subsets. The high-frequency rules are taken as the relevant rules to this task and the lower-frequency rules as irrelevant rules. In this way, one can cope with abundance and redundancy of rules before the subsets are assigned weights. For this reason, a new algorithm of rule selection is also constructed to optimize the assignment of weights.

The rest of this article is organized as follows. In the second section, a new model of mining very large transaction databases is presented, and in the third section, a rule selection algorithm of the model is described. Finally, the experiments are summarized in the last section.

WEIGHT AGGREGATION

When databases are too large (e.g., with terabytes of data) to be processed at one time for efficiency or space reasons, splitting them into subsets for processing is necessary. Since the number of itemsets in a data subset is a combinatorial amount and each of them may be a large itemset in the original database, data mining results from these subsets can possibly be larger in size than the data in the original database. Therefore, aggregating results from subsets is essential in large scale data mining from partitioned data subsets. To solve this problem, a weighting model is proposed in this section. The model is illustrated in Figure 1.

In this model, a given database VLDB is first partitioned into several subsets, SDB_i $(1 \leq i \leq n)$ according to allowed resources or requirements. Second, each subset SDB_i is mined, and all mined rules are stored in RS_i. According to the frequency of each rule appearing in RS_i $(1 \leq i \leq n)$, a

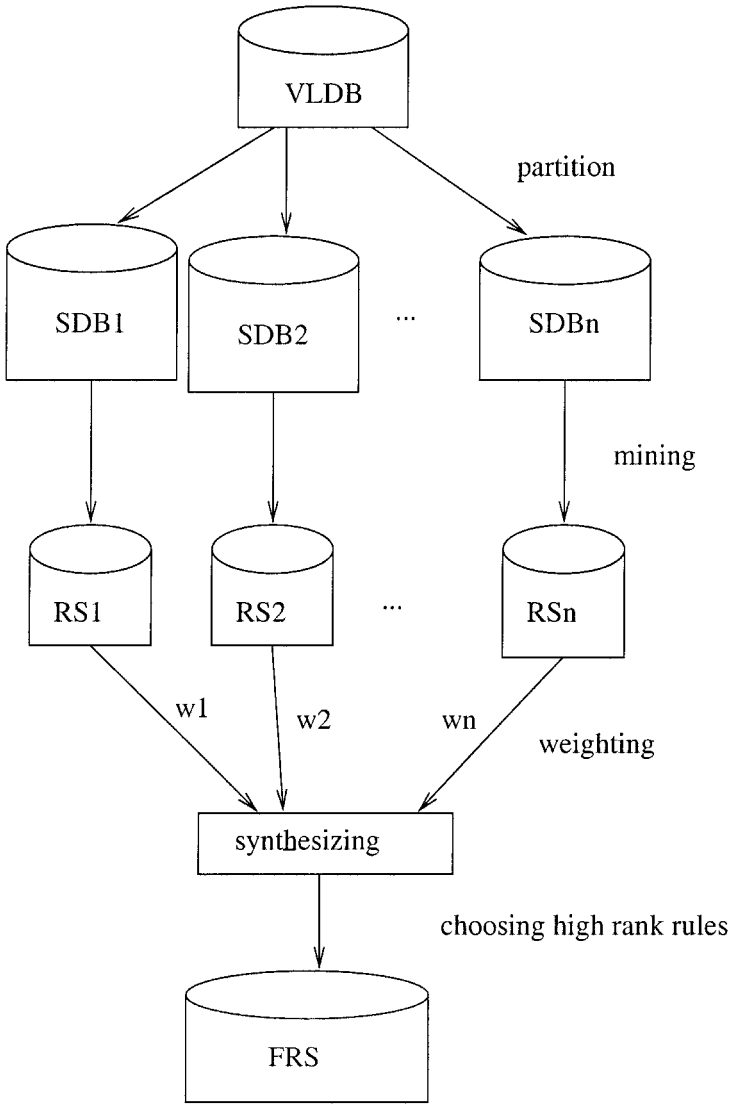

FIGURE 1. Mining very large databases. VLDB- a very large database to be mined; SDB_i $(1 \leq i \leq n)$-partitioned subsets of VLDB; RS_i- rules mined from SDB_i, where $1 \leq i \leq n$; synthesizing- a weighting procedure; FRS- weighted rules with high ranks.

weight can be assigned to each rule. Each subset SDB_i is also assigned a weight based on evidence that it supports high-frequency rules. Third, one can aggregate all rules by weighting. Finally, high rank rules are selected as the output.

Let $D_1, D_2, \ldots, D_m$ be m subsets of database D, S_i the set of association rules from D_i (i = 1, 2, ..., m), $S = \{S_1, S_2, \ldots, S_m\}$, and $R_1, R_2, \ldots, R_n$ be all rules in S. Suppose $w_1, w_2, \ldots, w_m$ are the weights of $D_1, D_2, \ldots, D_m$, respec-

tively. For a given rule $X \rightarrow Y$ in S, the aggregation is defined as follows:

$$p_w(X \cup Y) = w_1 * p_1(X \cup Y) + w_2 * p_2(X \cup Y) + \cdots + w_m * p_m(X \cup Y),$$

$$conf_w(X \rightarrow Y) = w_1 * conf_1(X \rightarrow Y) + w_2 * conf_2(X \rightarrow Y) + \cdots + w_m * conf_m(X \rightarrow Y),$$

where $p_i(X \cup Y)$ and $conf_i(X \rightarrow Y)$ are the support and confidence of $X \rightarrow Y$ in subset D_i $(1 \leq i \leq m)$.

As mentioned before, the aggregation of results in the weighting model is generally straightforward once all weights are reasonably assigned. To assign weights, Good's idea (1950) is first used to determine weights in this section, and then the weight aggregating algorithm is described.

Weight of Evidence

Good (1950) defines the weight in favor of a hypothesis, H, provided by evidence, E: the weight of evidence is calculated in terms of the ratio of the likelihoods. In this definition, he defines a concept "almost as important as that of probability itself." Good elucidates simple, natural desiderata for the formalization of the notion of weight of evidence, including an "additive property." This property states that the weight in favor of a hypothesis provided by two pieces of evidence is equal to the weight provided by the first piece of evidence, plus the weight provided by the second piece of evidence, conditioned on one having previously observed the first. Starting from these desiderata, Good is able to show that, up to a constant factor, the weight of evidence must take the form given in the definition of weight. It is an attractive scale because weights accumulate additively; it is also attractive because the entire range from $-\infty$ to $+\infty$ is used.

It is obvious that the weight of evidence can be used to aggregate the rules mined from the subsets of a given database. For convenience, weights are generally normalized into intervals [0, 1] in the following account.

Solving Weights

In order to aggregate association rules from the subsets of a given database, one needs to determine the weight for each partitioned subset. In our opinion, the weight of a subset is determined by the evidence that it supports high-frequency rules.

Let $D_1, D_2, \ldots, D_m$ be m subsets of database D, S_i the set of association rules from $D_i (i = 1, 2, \ldots, m)$, and $S = \{S_1, S_2, \ldots, S_m\}$. Intuitively, if a rule

$X \rightarrow Y$ has a high frequency in S, it would be assigned a large weight according to Good's idea, and $X \rightarrow Y$ has a high possibility to be synthesized as a useful rule. In other words, the more the number of subsets that contain the same rule, the larger the belief of the rule should be (or the larger the weight of the rule should be). This idea is illustrated by the following example.

Let D_1, D_2, D_3 be the three subsets of database D, minsupp = 0.2, minconf = 0.3, and the rules mined from three subsets are as follows:

1. S_1 the set of association rules from subset D1:
   $A \wedge B \rightarrow C$ with supp = 0.4, conf = 0.72; (R1)
   $A \rightarrow D$ with supp = 0.3, conf = 0.64; (R2)
   $B \rightarrow E$ with supp = 0.34, conf = 0.7; (R3)
2. S_2 the set of association rules from subset D2:
   $B \rightarrow C$ with supp = 0.45, conf = 0.87; (R4)
   $A \rightarrow D$ with supp = 0.36, conf = 0.7;
   $B \rightarrow E$ with supp = 0.4, conf = 0.6;
3. S_3 the set of association rules from subset D3:
   $A \wedge B \rightarrow C$ with supp = 0.5, conf = 0.82;
   $A \rightarrow D$ with supp = 0.25, conf = 0.62.

From the above data, two subsets support rule R_1, three subsets support rule R_2, two subsets support rule R_3, and one subset supports rule R_4. Following Good's weight of evidence, one can use the frequency of a rule as its weight. After normalization, the weights are assigned as follows: $w_{R1} = 0.25$, $w_{R2} = 0.375$, $w_{R3} = 0.25$, and $w_{R4} = 0.125$.

One has seen that rule R_2 has the highest frequency and the highest weight; and rule R_4 has the lowest frequency and the lowest weight. Let D_1, D_2, ..., D_m be the m subsets of database, D, S_i the set of association rules from D_i (i = 1, 2, ..., m), $S = \{S_1, S_2, \ldots, S_m\}$, and $R_1, R_2, \ldots, R_n$ be all rules in S. The weight of R_i is defined as follows:

$$\omega_{R_i} = \frac{\text{frequency}(R_i)}{\sum_{j=1}^{n} \text{frequency}(R_j)}$$

where, i = 1, 2, ..., n.

In the meantime, if a data subset supports a larger number of high-frequency rules, the weight of the subset should also be higher. If the rules from a subset are rarely present in other subsets, the subset would be assigned a lower weight. To implement this argument, one can use the sum of the multiplications of the rules' weights and their frequencies. For the above mined rules, one has $w_{D1} = 2 * 0.25 + 3 * 0.375 + 2 * 0.25 = 2.125$, $w_{D2} = 2$, and $w_{D3} = 1.625$. After normalization, the weights of these subsets are assigned as $w_{D1} = 0.3695$, $w_{D2} = 0.348$, and $w_{D3} = 0.2825$.

One has seen that subset D_1 supports the most high-weight rules, and accordingly, it has the highest weight; and subset D_3 supports the smallest high-weight rules and it has the lowest weight.

Let $D_1, D_2, \ldots, D_m$ be m subsets of D, S_i the set of association rules from D_i ($i = 1, 2, \ldots, m$), $S = \{S_1, S_2, \ldots, S_m\}$, and $R_1, R_2, \ldots, R_n$ be all rules in S. The weight of D_i is defined as follows:

$$w_{D_i} = \frac{\sum_{R_k \in S_i} w_{R_k} * \text{frequency}(R_k)}{\sum_{j=1}^{m} \sum_{R_h \in S_j} w_{R_j} * \text{frequency}(R_j)},$$

where, $i = 1, 2, \ldots, m$.

After the weights have been assigned to different data subsets, one can aggregate the association rules from these subsets. The aggregation process is demonstrated as follows.

Example 1. For rule R_1: $A \wedge B \rightarrow C$,

$$\begin{aligned} p(A \cup B \cup C) &= w_{D1} * p_1(A \cup B \cup C) + w_{D3} * p_3(A \cup B \cup C) \\ &= 0.3695 * 0.4 + 0.2825 * 0.5 = 0.28905, \\ \text{conf}(A \wedge B \rightarrow C) &= w_{D1} * \text{conf}_1(A \wedge B \rightarrow C) + w_{D3} * \text{conf}_3(A \wedge B \rightarrow C) \\ &= 0.3695 * 0.72 + 0.2825 * 0.82 = 0.49769. \end{aligned}$$

For rule R_2: $A \rightarrow D$,

$$\begin{aligned} p(A \cup D) &= w_{D1} * p_1(A \cup D) + w_{D2} * p_2(A \cup D) + w_{D3} * p_3(A \cup D) \\ &= 0.3695 * 0.3 + 0.348 * 0.36 + 0.2825 * 0.25 = 0.306755, \\ \text{conf}(A \rightarrow D) &= w_{D1} * \text{conf}_1(A \rightarrow D) + w_{D2} * \text{conf}_2(A \rightarrow D) \\ &\quad + w_{D3} * \text{conf}_3(A \rightarrow D) \\ &= 0.3685 * 0.64 + 0.348 * 0.7 + 0.2825 * 0.62 = 0.68043. \end{aligned}$$

For rule R_3: $B \rightarrow E$,

$$\begin{aligned} p(B \cup E) &= w_{D1} * p_1(B \cup E) + w_{D2} * p_2(B \cup E) \\ &= 0.3695 * 0.34 + 0.348 * 0.4 = 0.26483, \\ \text{conf}(B \rightarrow E) &= w_{D1} * \text{conf}_1(B \rightarrow E) + w_{D2} * \text{conf}_2(B \rightarrow E) \\ &= 0.3695 * 0.7 + 0.348 * 0.6 = 0.46745. \end{aligned}$$

For rule R_4: $B \rightarrow C$,

$$\begin{aligned} p(B \cup C) &= w_{D2} * p_2(B \cup C) = 0.348 * 0.45 = 0.1566, \\ \text{conf}(B \rightarrow C) &= w_{D2} * \text{conf}_2(B \rightarrow C) = 0.348 * 0.87 = 0.30276. \end{aligned}$$

The ranking of the above rules is R_2, R_1, R_3, and R_4 by their supports. According to this ranking, one can select high-rank rules after the minimum support and minimum confidence.

Algorithm Design

Let D be a given very large transaction database, and minsupp, minconf the threshold values given by the user. Our weighting algorithm for mining association rules in D is designed as follows.

Algorithm 1. Weight aggregation
Input: D: a very large database; minsupp, minconf: threshold values;
Output: S: a set of association rules;
(1) partition D into several subsets;
(2) mine each subset;
(3) assign a weight to each subset;
(4) aggregate all rules by weighting;
(5) rank the rules;
(6) select high-rank rules to S, which have both support $\geq$ minsupp and confidence $\geq$ minconf;
(7) output S;
(8) end all.

RULES SELECTION

Using the above model, one can assign a higher weight to a data subset of a given database by the evidence that the subset supports more high-frequency rules, and a lower weight to a subset that supports less high-frequency rules. However, this model can be optimized by rules selection. The following two examples illustrate rule selection.

Example 2. Let $D_1, D_2, \ldots, D_{11}$ be the 11 subsets of D, S_i the set of association rules from D_i ($i = 1, 2, \ldots, 11$), $S_i = \{R_1\}$ when $i = 1, 2, \ldots, 10$, and $S_{11} = \{R_2, R_3, \ldots, R_{11}\}$. Then we have

$$w_{R1} = \frac{\text{frequency}(R_1)}{\sum_{j=1}^{11} \text{frequency}(R_j)} = \frac{10}{10 + \sum_{j=1}^{10} 1} = 0.5,$$

$$w_{R_i} = \frac{\text{frequency}(R_i)}{\sum_{je1}^{11} \text{frequency}(R_j)} = \frac{1}{10 + \sum_{j=1}^{10} 1} = 0.05,$$

where i = 2, 3, ..., 11.

So,

$$w_{D_i} = \frac{\sum_{R_k \in S_i} \text{frequency}(R_k) * w_{R_k}}{\sum_{j=1}^{m} \sum_{r_h \in S_j} \text{frequency}(R_h) * w_{R_h}} = \frac{10 * 0.5}{\sum_{j=1}^{10} 10 * 0.5 + \sum_{j=1}^{10} 1 * 0.05} = 0.099,$$

where i = 1, 2, ..., 10.

$$w_{D11} = \frac{\sum_{R_k \in S_{11}} \text{frequency}(R_k) * w_{R_h}}{\sum_{j=1}^{m} \sum_{R_h \in S_j} \text{frequency}(R_h) * w_{R_h}} = \frac{\sum_{j=1}^{10} 1 * 0.05}{\sum_{j=1}^{10} 10 * 0.5 + \sum_{j=1}^{10} 1 * 0.05} = 0.01.$$

One has seen that although S_{11} has 10 rules, w_{D11} is still very low due to the fact that S_{11} doesn't contain high-frequency rules. If $S_{11} = \{R_2, R_3, \cdots, R_{91}\}$, then one has $w_{R1} = 0.1$, and $w_{Ri} = 0.01$, where i = 2, 3, ..., 91. So, $w_{Di} = 0.09174$ ($1 \leq i \leq 10$), and $w_{D11} = 0.0826$.

In this case, w_{D11} becomes higher. Although $_{D11}$ cannot cause rules R_i ($2 \leq i \leq 91$) to become valid rules in synthesis, the support and confidence of R_1 are slightly weakened by w_{D11}. The larger the number of rules in S_{11}, the more the other rules are weakened.

R_1 is the object extracted from $S = \{S_1, S_2, \ldots, S_{11}\}$ in Example 2. The rules with lower frequency (for example, less than 2) can be taken as noise. For efficiency purposes, this noise could be wiped out before the subsets are assigned weights. To eliminate this noise, because the goal is to extract high-frequency rules from different subsets, one can take the high-frequency rules as the relevant rules and the lower-frequency rules as irrelevant rules. In this way, one can cope with abundance and redundancy of rules before the subsets are assigned weights. For this reason, one constructs a new algorithm of rule selection for mining large frequent itemsets from transaction databases in this section.

Let $D_1, D_2, \ldots, D_m$ be m subsets of D, S_i the set of association rules from D_i(i = 1, 2, ..., m), $S = \{S_1, S_2, \ldots, S_m\}$. Assume there are N rules in S. Rule selection is to select a minimum set of M high-frequency rules from S where $M \leq N$ such that all association rules that are extracted are still included in S. In other words, those rules whose frequencies are less than a specified threshold, called minimum frequency (γ), are deleted from each transaction in S. This procedure is implemented as follows.

Procedure 1. RuleSelection (S: a set of transactions with rules as their items);
Input: γ-allowed minimal frequency, S-set of N rules;
Output: S-set of M rules whose frequencies $\geq \gamma$;

```
for each rule R in S do
  if (the frequency of R in S is less than γ)
    for any transaction t in S
      if (R is present in t)
        delete R from t;
    end for
end for
```

The above procedure can be used in Algorithm 1 to improve its efficiency, after step (2) and before step (3). One now shows the impact of rule selection on weights.

Let $D_1, D_2, \ldots, D_{10}$ be the 10 subsets of a given database, S_i the set of association rules from D_i ($i = 1, 2, \ldots, 10$) $S_i = \{R_1\}$ when $i = 1, 2, \ldots, 9$, and $S_{10} = \{R_1, R_2, \ldots, R_{11}\}$. Then one has $w_{R1} = 0.5$, and $w_{Ri} = 0.05$, where $i = 2, 3, \ldots, 11$. So, $w_{Di} = 0.099$ ($1 \leq i \leq 9$), and $w_{D11} = 0.109$.

Because the frequency of R_i is 1 ($2 \leq i \leq 11$), rules R_i ($2 \leq i \leq 11$) can all be wiped out as noise. After wiping out the noise, one has $w_{Di} = 0.1$, where $i = 1, 2, \ldots, 10$. The errors of databases D_i ($1 \leq i \leq 9$) are all 0.001, and the error of D_{10} is 0.009.

EXPERIMENTS AND CONCLUSIONS

To evaluate the effectiveness of the above aggregation model, several experiments have been performed. Oracle 8.0.3 was used for database management, and the aggregation model was implemented on Sun Sparc using Java. The databases used are three market transaction databases from the synthetic classification data sets on the Internet (http://www.kdnuggets.com/). The main properties of the three databases are as follows. There are $|R| = 1000$ attributes in each database. The average number T of attributes per row is 5, 10, and 20, respectively. The number $|r|$ of rows is approximately 100,000 in each database. The average size I of maximal frequent sets is 2, 4, and 6, respectively, and the number of partitions ns for each database is 5, 8, and 10. Table 1 summarizes these parameters.

These databases were first mined with Algorithm 1, and then the rule selection procedure was used in the second mining. The results have shown that the first 20 association rules mined in the weighting model are consistent with the Apriori algorithm when minsupp = 0.01, minconf = 0.65, and all rules were ranked by their supports. This model was significantly faster than the Apriori algorithm: the execution time was approximately one-tenth of

TABLE 1 Data Characteristics

Database Name	|R|	T	I	|r|	ns
T5.I2.D100K	1000	5	2	100051	5
T10.I4.D100K	1000	10	4	98749	8
T20.I6.D100K	1000	20	6	99408	10

the time of the Apriori algorithm when the rule selection procedure was used with minimum frequencies 1, 2, and 3 for datasets T5.I2.D100K, T10.I4.D100K, and T20.I6.D100K, respectively.

When databases are very large and the resources are bounded, existing data mining models do not work well because they require multiple passes over the original database. For this reason, a weighting model has been proposed to deal with very large databases in this paper. In this model, a given database is first partitioned into several subsets according to allowed resources or requirements. Second, each subset is mined. Third, all rules can be aggregated by weighting. Finally, some high-rank rules are selected as the output. In particular, the efficiency of this model is improved by calling a new procedure for rule selection.

REFERENCES

Agrawal, R., T. Imielinski, and A. Swami. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 207–216. Washington DC.: ACM Press.

Brin, S., R. Motwani, and C. Silverstein. 1997. Beyond market baskets: Generalizing association rules to correlations. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, R265–276. Tucson, AZ: ACM Press.

Good, I. 1950. *Probability and the weighting of evidence*. London: Charles Griffin.

Hidber, C. 1999. Online association rules mining. In: *Proceedings of the ACM SIGMOD Conference on Management of Data*, Philadelphia, PA: ACM Press.

Park, J. S., M. S. Chen, and P. S. Yu. 1995. An effective hash based algorithm for mining association rules. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, San Jose, California: ACM Press.

Srikant, R., and R. Agrawal. 1997. Mining generalized association rules. *Future Generation Computer Systems* 13: 161–180.

Toivonen, H. 1996: Sampling large databases for association rules. In *Proceedings of the 22nd VLDB Conference*, 134–145. Bombay, India: Morgan Kaufmann.