ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2009/03(04)-0392-13
DOI:10.3778/j.issn.1673-9418.2009.04.006

E-mail:fcst@public2.bta.net.cn
http://www.ceaj.org
Tel:+86-10-51616056

MPSQAR: Mining Quantitative Association Rules without Loss of Semantics*

ZENG Chunqiu+, TANG Changjie, LI Chuan, DUAN Lei
School of Computer Science, Sichuan University, Chengdu 610065, China
+ Corresponding author: E-mail: zengchunqiu@cs.scu.edu.cn

MPSQAR:无损语义的量化关联规则挖掘算法*

曾春秋+,唐常杰,李 川,段 磊
四川大学 计算机学院,成都 610065

摘 要:在挖掘量化关联规则的过程中,由于对量化值的划分,将产生语义损失。为避免这种情况,提出基于无损语义的算法 MPSQAR 来处理量化关联规则的挖掘。主要工作包括:(1)提出规泛化量化值的新方法;(2)提出反映属性值分布的属性权重设计方法;(3)扩展加权关联规则模型以处理量化关联规则,避免量化值的划分;(4)提出挖掘传统布尔关联规则和量化关联规则的集成方法;实验表明算法 MPSQAR 的有效性和时间消耗随时间趋势呈线性增长。

关键词:量化关联规则;MPSQAR 算法;语义信息损失
文献标识码:A **中图分类号:**TP301

ZENG Chunqiu, TANG Changjie, LI Chuan, et al. MPSQAR: Mining quantitative association rules without loss of semantics. Journal of Frontiers of Computer Science and Technology, 2009,3(4):392-404.

Abstract: During the process of mining quantitative association rules, the semantics may be lost due to the discretization of quantitative values. To avoid the loss of semantic information, a novel algorithm, MPSQAR (mining preserving semantic quantitative association rule), is proposed to handle the quantitative association rules mining. The main contributions include: (1) Propose a new method to normalize the quantitative values; (2) Propose a

* The National Natural Science Foundation of China under Grant No.600773169 (国家自然科学基金); the National Great Project of Scientific and Technical Supporting Programs Funded by Ministry of Science & Technology of China During the 11th Five-year Plan under Grant No.2006BAI05A01 (国家"十一五"科技支撑计划重大项目资助).
Received 2008-11, Accepted 2009-03.

method to assign a weight for each attribute to reflect the values distribution; (3) Extend the weight-based association model to tackle the quantitative values in association rules without partition; (4) Design a integrated and uniform method to mine the traditional Boolean association rules and quantitative association rules; Experiments show the effectiveness and linear scalability of the new method on time consuming.

Key words: quantitative association rule; MPSQAR algorithm; loss of semantic information

1 Introduction

Association rule is an important expression of knowledge to reveal implicit relationships among the items present in large number of transactions[1]. Mining association rule with binary values, called binary association rule mining, is well studied[2-4]. For association rules with categorical values, each attribute, with several specific values without semantic order, can be converted into several binary attributes for each categorical value[5-8]. Association rule with quantitative values is called quantitative association rule[7].

The previous researches mainly find the association rules by partitioning the quantitative domain and transforming the problem into several binary ones[9-12]. Also, many studies have been made on the methods to divide the quantitative values[7,13-14], such as equal-width binning, equal-frequency binning and clustering-based binning[5], and fuzzy sets theory is employed to divide values into different bins[8]. However, the mining results just can reflect the associations among bins of different attributes rather than the associations among all the attributes[15]. Min-apriori[15] processes the quantitative data directly by normalizing the quantitative value. All values in each column are added up to 1.0. Let T be a table used to represent a data set and $T(i,j)$ be the responding value of j-th item in i-th transaction and let $T(i)$ be the i-th transaction in data set T. According to[15], the $T(i,j)$ can be normalized into $T_n(i,j)$ as the following way:

$$T_n(i,j)=\frac{T(i,j)}{\sum_{i=1}^{|T|}T(i,j)} \tag{1}$$

Where $|T|$ is the size of data set. According to Equation (1), it is easy to see that $T_n(i,j)$ ranges from 0.0 to 1.0.

Finding frequent item sets and extracting interesting association rules from frequent item sets are key phases in mining association rules[1-4,15]. An item set is frequent if its support is larger than the minimum support. To extract interesting rules from frequent item sets, a rule is interesting only when its confidence is larger than the minimum confidence. Both minimum support and minimum confidence are user-specified values. In particular, for binary values, Apriori is one of the famous algorithms to mine association rules. For Apriori algorithm, the support of an item set X is calculated by:

$$support(X)=\frac{|\{t|t\in T\wedge X\subseteq t\}|}{|T|} \tag{2}$$

where t is one of transactions set T and $|T|$ is the size of the transactions set.

For Min-apriori in [15], after the normalization of the data set, the support of an item set X, denoted as $support(X)$, is defined as follows:

$$support(X)=\sum_{i=1}^{|T|}\min\{T_n(i,j)|j\in X\} \tag{3}$$

In Equation (3), the support of X in Min-apriori is defined as the sum of all the minimum $T_n(i,j)$ values of each transaction in data set. Min-apriori can keep the quantitative semantics during the phase of association rules mining. The larger the attribute value, the more

contributions of the attribute entry to the support.

In Table 1, the data set contains five transactions and six items, $I=\{A,B,C,D,E,F\}$. Following the Equation (1), the data set is normalized (Table 2). By Equation (3), for each item $i \in I$, $support(\{i\})=1.0$, thus the item set $\{i\}$ containing single item is frequent. This is called side effect. That does not show the truth: $\{i\}$ occurs rarely.

Table 1 A data set containing 5 transactions and $I=\{A,B,C,D,E,F\}$

表 1 由 5 个事务构成的数据集,项集 $I=\{A,B,C,D,E,F\}$

TID	A	B	C	D	E	F
TID_1	10	5	1	0	0	10
TID_2	2	5	1	0	0	0
TID_3	0	0	0	1	2	2
TID_4	1	5	1	0	0	0
TID_5	0	0	0	0	0	1

Table 2 Data set after normalization

表 2 正规化的数据集

TID	A	B	C	D	E	F
TID_1	0.77	0.33	0.33	0.0	0.0	0.77
TID_2	0.15	0.33	0.33	0.0	0.0	0.00
TID_3	0.00	0.00	0.00	1.0	1.0	0.15
TID_4	0.08	0.33	0.33	0.0	0.0	0.00
TID_5	0.00	0.00	0.00	0.0	0.0	0.08
Total	1.00	1.00	1.00	1.0	1.0	1.00

To solve the previous problems, the main contributions of this paper include:

(1) Introduce a new normalization method to eliminate the side effect;

(2) Employ a weight measure for each attribute to show the different distribution;

(3) Propose MPSQAR algorithm to mine quantitative association rules by extending the weighted association rule mining model[1] which introduces weight artificially according to the interest for each attribute and focuses on binary values.

The rest of this paper is organized as follows. Section 2 describes the new way to normalize the quantitative values. Section 3 introduces weight according the variance of the values distribution for each attribute. Section 4 presents the MPSQAR algorithm for mining quantitative association rules by incorporating weight into Minapriori[15] algorithm and revising weighted association rule mining model[16]. Section 5 gives experiments to show the effective and scalable performance of MPSQAR algorithm. And Section 6 concludes the paper.

2 Quantitative Values Normalization

Consider the data set with traditional binary values in Table 3. Traditional support[1-2] of an item set can be calculated by Equation (2) which is illustrated in the following example.

Table 3 Data set with binary values

表 3 只含布尔值的数据集

TID	A	B	C	D	E	F
TID_1	1	1	1	0	0	1
TID_2	1	1	1	0	0	0
TID_3	0	0	0	1	1	1
TID_4	1	1	1	0	0	0
TID_5	0	0	0	0	0	1

Example 1 Given item set $\{A\}$, then $support(\{A\})=|\{t|\ t \in T \wedge \{A\} \in t\}|/|T|=3/5=0.6$, where $|T|$ is the count of transactions. Normalizing the data set by Equation (1) and then calculating the support of item set $\{A\}$ by (3), we get $support(\{A\})=1/(1+1+0+1+0)+1/(1+1+0+1+0)+1/(1+1+0+1+0)=0.33+0.33+0.33=1.0$. Both supports are quite different from each other and the side effect occurs again in the latter one.

In order to eliminate the side effect and unify both the binary and quantitative situations, we handle a

specific attribute values according to the following steps:

(1) Estimate the most possible non-zero value occurring in the attribute column.

(2) Calculate the most possible sum of the attribute values if all the entries present in the attribute column with non-zero value.

(3) Employ the most possible sum to normalize the non-zero values of the attribute.

Especially, for mining traditional association rules with binary values referring to the Table 3, 1 is supposed to be most possible nonzero value and the most possible sum of the attribute values is $|T|$. According to Equation (1), each $T(i,j)=1$ is normalized into $T_n(i,j)=1/|T|$. Naturally, by Equation (3), $support(\{A\})=1/(1+1+1+1+1)+1/(1+1+1+1+1)+1/(1+1+1+1+1)=0.6$ with the same result as the one by Equation (2). To describe our algorithms clearly, we introduce following new concepts.

Definition 1 Let v be the most possible nonzero value to occur in the j-th attribute. Given an entry for one attribute column in the data set table whose original value is 0. Then v is called expecting value filled (EVF), and defined as follows:

$$EVF(j)=\sum_{i=1}^{|T|}T(i,j)\times \frac{1}{|\{T(i)|T(i,j)\neq 0,1\leqslant i\leqslant |T|\}|} \tag{4}$$

In Equation (4), the numerator is the sum of all the values for the specific j-th attribute in all the transactions. And the denominator is the count of transactions whose values for j-th attribute are nonzero.

Example 2 Considering the Table 1, EVF values of all the attributes are listed:

$EVF(1)=(10+2+0+1+0)/(1+1+1)=13/3$

$EVF(2)=(5+5+0+5+0)/(1+1+1)=5$

$EVF(3)=(1+1+0+1+0)/(1+1+1)=1$

$EVF(4)=(0+0+1+0+0)/1=1$

$EVF(5)=(0+0+2+0+0)/1=2$

$EVF(6)=(10+0+2+0+1)/(1+1+1)=13/3$

Especially, Consider binary values in Table 3. The responding EVF results are listed:

$EVF(1)=(1+1+0+1+0)/3=1$

$EVF(2)=(1+1+0+1+0)/3=1$

$EVF(3)=(1+1+0+1+0)/3=1$

$EVF(4)=(0+0+1+0+0)/1=1$

$EVF(5)=(0+0+1+0+0)/1=1$

$EVF(6)=(1+0+1+0+1)/3=1$

In binary values situation, all the EVF results for all the attributes are 1.

Definition 2 The normalization coefficient: Is a real number defined by formula (5):

$$NC(j)=\frac{1}{|T|\times EVF(j)} \tag{5}$$

Note that, intuitively, NC is the normalization of j-th attribute in the data set.

By the Definition 2, given value $T(i,j)$, let $T_n(i,j)$ be the value after normalization. Then the normalization can be described as the following equation:

$$T_n(i,j)=T(i,j)\times NC(j) \tag{6}$$

Example 3 Consider the Table 1 and Table 3. The results of normalization are shown in Table 4 and Table 5.

Table 4 The normalization result of Table 1 by NC

表 4 表 1 通过 NC 正规化的结果数据集

TID	A	B	C	D	E	F
TID_1	0.46	0.2	0.2	0.0	0.0	0.46
TID_2	0.09	0.2	0.2	0.0	0.0	0.00
TID_3	0.00	0.0	0.0	0.2	0.2	0.09
TID_4	0.05	0.2	0.2	0.0	0.0	0.00
TID_5	0.00	0.0	0.0	0.0	0.0	0.05
Total	0.60	0.6	0.6	0.2	0.2	0.60

Table 5 The normalization result of Table 3 by *NC*

表 5 表 3 通过 *NC* 正规化的结果数据集

TID	*A*	*B*	*C*	*D*	*E*	*F*
TID_1	0.2	0.2	0.2	0.0	0.0	0.2
TID_2	0.2	0.2	0.2	0.0	0.0	0.0
TID_3	0.0	0.0	0.0	0.2	0.2	0.2
TID_4	0.2	0.2	0.2	0.0	0.0	0.0
TID_5	0.0	0.0	0.0	0.0	0.0	0.2
Total	0.6	0.6	0.6	0.2	0.2	0.6

Calculate the support for item set $\{A\}$ by Equation (3) as Table 4, then we get $support(\{A\})=0.46+0.09+0.05=0.6$. Thus the side effect does not occur if the size of the item set is small as shown in Table 2. Suppose minimum support $minsup=0.3$, calculate traditional support with Equation (2), we get $support(\{A,F\})=0.2$. Thus $\{A,F\}$ is not frequent. Since $support(\{A,F\})=0.46$ by Equation (3), $\{A,F\}$ is frequent. Especially, considering the binary values situation, by Table 5 and Equation (3), the supports of all the item sets are the same as the traditional supports calculated with Equation (2).

Lemma 1 Let $tsupport(X)$ be the support of X calculated by Equation (2) and $nsupport(X)$ be the support of X calculated by Equation (3) after normalization with *NC*. Given an item set X of a data set T. Assume that all the items in T are binary attributes. Then $tsupport(X)=nsupport(X)$.

Proof Let $T(i,j)$ be the value of the j-th attribute in the i-th transaction, $T_n(i,j)$ be the value of normalizing $T(i,j)$, $T(i)$ be the i-th transaction of the data set T. Firstly, $tsupport(X)=|\{t|t\in T\wedge X\subseteq t\}|/|T|=|\{T(i)|1\leqslant i\leqslant |T|\wedge X\subseteq T(i)\}|/|T|$. Since if $X\subseteq T(i)$, then $\min\{T(i,j)|j\in X\}=1$, and if $X\subseteq T(i)$, then $\min\{T(i,j)|j\in X\}=0$. Thus:

$$tsupport(X)=\sum_{i=1}^{|T|}\frac{\min\{T(i,j)|j\in X\wedge X\subseteq T(i)\}}{|T|}+\sum_{i=1}^{|T|}\frac{\min\{T(i,j)|j\in X\wedge X\subseteq T(i)\}}{|T|}\Rightarrow$$

$$tsupport(X)=\sum_{i=1}^{|T|}\frac{\min\{T(i,j)|j\in X\}}{|T|}$$

On the other hand:

$$nsupport(X)=\sum_{i=1}^{|T|}\min\{T_n(i,j)|j\in X\}$$

According to Equation (5) and (6), we can draw that:

$$nsupport(X)=\sum_{i=1}^{|T|}\min\{T(i,j)\times NC(j)|j\in X\}=\sum_{i=1}^{|T|}\min\{\frac{T(i,j)}{|T|\times EVF(j)}|j\in X\}$$

As described in Example 2, all the $EVF(j)=1$ in the binary values situation, therefore:

$$\sum_{i=1}^{|T|}\min\{T(i,j)/(|T|)|j\in X\}=\sum_{i=1}^{|T|}\min\{T(i,j)|j\in X\}|T|/tsupport(X)$$

So $tsupport(X)$ is equal to $nsupport(X)$. □

Lemma 1 shows that the normalization method unifies support definitions in both traditional binary values situation and quantitative values situation.

3 Incorporate Weight into Quantitative Association Rules

3.1 Introducing Weight

In previous two sections, Equation (1) is applied for normalization Min-apriori[15], and Equation (6) is for normalization without side effect. It unifies the support definitions in both binary and quantitative situations. However, both equations ignore the distribution of the values in attribute. By careful consideration on the weight of quantitative association rules, we have the follwoing observations.

Observation 1 In Table 1, the values of attribute A and attribute B are distributed quite differently. However, after normalizing the data by Equation (6) into Table 4, and calculating support by Equation (3), $support(\{A\})$ is same as $support(\{B\})$ although the dis-

tributions of A and B are quite different especially when the size of item set is not large enough.

Observation 2 In Table 1, attribute C always occurs with 1 or 0. So C is supposed to be a binary attribute. Comparing A with C in Table 4, it is obvious that $support(\{A\})$ is equal to $support(\{C\})$. So Equation (3) can not reflect the difference between A and C. And a reasonable result that $support(\{A\})$ is greater than $support(\{C\})$ is expected.

Based on the observations above, it is worthwhile to incorporate the distributions of different attributes into the way of calculating support.

Observation 3 In Table 1, attribute B always occurs with five or zero in data set. Thus it should also be viewed as a binary attribute. As a result, that $support(\{B\})$ equals to $support(\{C\})$ is considered to be reasonable.

In order to reflect the distribution of each attribute described in Observation 1 and 2, and keep the property in Observation 3, a weight should be introduced for each attribute in the method of calculating support.

For the convenient in later discussion, we denote the array containing all the nonzero attribute value as $NAVA$.

Example 4 According to the description above, the following are obvious for Table 1:

$NAVA(A)=NAVA(1)=\{10,2,1\}$

$NAVA(B)=NAVA(2)=\{5,5,5\}$

$NAVA(C)=NAVA(3)=\{1,1,1\}$

$NAVA(D)=NAVA(4)=\{1\}$

$NAVA(E)=NAVA(5)=\{2\}$

$NAVA(F)=NAVA(6)=\{10,2,1\}$

Considering Definition 1 and Example 2, it can be easily found that EVF value of the j-th attribute is the mean value of $NAVA(j)$.

In order to reflect the variance of the distribution for a specific attribute, absolute deviation is employed and a new concept is given as follows.

Definition 3 Let $NAVA(i,j)$ be the i-th value in the $NAVA(j)$, and $|NAVA(j)|$ be the size of $NAVA(j)$ array. Then the relative diversity value of $NAVA(j)$, denoted as v, is said to be the variance factor of the j-th attribute, abbreviated as VF, defined as:

$$VF(j)=\sum_{i=1}^{|NAVA(j)|}|NAVA(i,j)-EVF(j)|\times\frac{1}{|NAVA(i,j)|\times EVF(j)} \tag{7}$$

Note that in Definition 3, $VF(j)$ reflects the variance of the j-th attribute relative to $EVF(j)$, that is the expecting value of $NAVA(j)$ for the j-th attribute.

Lemma 2 Given a data set T, let $T(i,j)\geqslant 0$ be the value of the j-th attribute in the i-th transaction of T, then $VF(j)\in[0,2]$ if and only if each $NAVA(i,j)=EVF(j)$, $VF(j)=0$.

Proof [**Preparation**] Since each $NAVA(i,j)\neq 0$ and $T(i,j)\geqslant 0$, thus $NAVA(i,j)\geqslant 0$. By the Example 4, $EVF(j)$ is the mean of all the values in $NAVA(j)$, so $EVF(j)>0$. By Equation (7), it is inferred easily that $VF(j)\geqslant 0$. On the other hand, given another two arrays $left(j)$ and $right(j)$, let $left(j)$ contains all the $NAVA(i,j)<EVF(j)$, $right(j)$ contains all the $NAVA(i,j)\geqslant EVF(j)$, then $|left(j)|+|right(j)|=|NAVA(j)|$ and $\sum_{i=1}^{|left(j)|}(EVF(j)-left(i,j))=\sum_{i=1}^{|left(j)|}(right(i,j)-EVF(j))$ since $EVF(j)$ is the mean of all the $NAVA(i,j)$. And we can draw that: $\sum_{i=1}^{|NAVA(j)|}|NAVA(i,j)-EVF(j)|=\sum_{i=1}^{|left(j)|}|left(i,j)-EVF(j)|+\sum_{i=1}^{|right(j)|}|right(i,j)-EVF(j)|\Rightarrow |NAVA(i,j)-EVF(j)|=2\times\sum_{i=1}^{|left(j)|}(EVF(j)-left(i,j))\leqslant 2\times|NAVA(j)|\times EVF(j)$, therefore:

$$VF(j)=\frac{\sum_{i=1}^{|NAVA(j)|}|NAVA(i,j)-EVF(j)|}{|NAVA(j)|\times EVF(j)}\leqslant \frac{2\times|NAVA(j)|\times EVF(j)}{|NAVA(j)|\times EVF(j)}=2$$

[**Sufficiency**] From the proof above, when each $NAVA(i,j)\rightarrow EVF(j)$, then numerator$\rightarrow$0. That is, $NAVA(i,j)=EVF(j)$ implied $VF(j)=0$. And the more the values of the j-th attribute vary, the greater the $VF(j)$ is.

[**Necessity**] Note that, if $VF(j)=0$, then the numerator of Equation (7) equals to 0. So $NAVA(i,j)=EVF(j)$. □

By Definition 3 and Lemma 1, given the specific j-th attribute, then the weight of the j-th attribute can be defined as follows:

$$weight(j)=1+\frac{VF(j)}{2} \tag{8}$$

Lemma 3 Given the specific j-th attribute, let $weight(j)$ be the weight of the j-th attribute as defines above. Then: (1) $weight(j)\in[1,2]$; (2) When each $NAVA(i,j)$ approaches $EVF(j)$, then $weight(j)$ approaches 1.

Proof It follows from Lemma 2 immediately. Note that, the more the values of the j-th attribute vary, the greater the $weight(j)$ is. Especially, if the j-th item is a binary attribute, then it is inferred easily that $VF(j)=0$, therefore, $weight(j)=1$.

Example 5 Consider Table 1. By Equation (8), all the weights of all the attributes list as follows: $weight(1)=1.0+0.44=1.44$; $weight(2)=1.0+0.0=1.0$; $weight(3)=1.0+0.0=1.0$; $weight(4)=1.0+0.0=1.0$; $weight(5)=1.0+0.0=1.0$; $weight(6)=1.0+0.44=1.44$; It is clear that $weight(1)$ and $weight(6)$ are the greatest due to their most variational distribution. And the rest attributes can be consider being binary attributes, so that all the weights are 1 is reasonable.

3.2 Modeling Weight

In order to incorporate weights of quantitative attributes into association rules, the definition of support in Equation (3) should be revised. To model the weight, weighted support and normalized weight support for the association rules are proposed in [16]. However, the traditional support does not work well for quantitative association rules. Thus, a new weighted support is proposed to meet the weighted quantitative association rules.

Let T be a data set, $T(i,j)$ be the value of the j-th attribute in the i-th transaction of T, and $T_n(i,j)$ be the value after $T(i,j)$ being normalized by Equation (6). We have:

Definition 4 Given an item set X for data set T and the j-th item attribute in the item set X, let $weight(j)$ be the weight of the j-th attribute, and let $wsupport(X)$ denote the weighted support. The weighted support is defined as follows:

$$wsupport(X)=\frac{1}{|X|}\sum_{j\in X}weigh(j)\times\sum_{i=1}^{|T|}\min\{T_n(i,j)|j\in X\} \tag{9}$$

Note that, as defined in [1,2], let $minsup$ be a user specified minimum support, if $wsupport(X)\geqslant minsup$, then X is a large (or frequent) item set.

Example 6 Consider Table 4. Suppose the minimum support $minsup=0.1$. Let $X=\{E,F\}$, then $support(X)=0.09$ by Equation (3), and X is not a large item set; on the other hand, $wsupport(X)=((1+1.44)/2)\times0.09=0.109\ 8$ and $wsupport(X)>0.1$, so X is considered as a large item set and if $|X|=k$, X is called large k-item set.

Lemma 4 Given a data set T, suppose all the items in T are binary attributes, and an item set X for data set T. Given the j-th item attribute in the item set X,

Let $weight(j)$ be the weight of the j-th attribute, and let $wsupport(X)$ denote the weighted support, $tsupport(X)$ denote the support of X calculated by Equation (2) and $nsupport(X)$ be the support of X defined by Equation (3). Then $tsupport(X)=nsupport(X)=wsupport(X)$.

Proof First, according to Lemma 1, since all the items are binary attributes

$tsupport(X)=nsupport(X)$

Second, consult to the Equation (9), then

$$wsupport(X)=\frac{1}{|X|}\sum_{j\in X} weight(j)\times nsupport(X)$$

Third, because of all the binary attributes, for the j-th attribute, $weight(j)=1$, so $wsupport(X)=nsupport(X)$.

The conclusion is $tsupport(X)=nsupport(X)=wsupport(X)$. This completes the proof. □

As shown in Lemma 4, the definition of weighted support can handle the support of item set in both data sets with binary and quantitative attributes. Thus, there is no loss of power in tackling the binary attribute; also it can handle the quantitative attribute with the ability of reflecting the distribution of attribute values directly.

Given two item sets X and Y, and $X\cap Y=\varnothing$, an association rule r can be defined in the form: $X\Rightarrow Y$. Let $wsupport(X)$ be the weighted support of X described in Definition 4 and $nsupport(X)$ be the support of X defined in Equation (3). Thus:

(1) The support of r is:

$support(r)=wsupport(X\cup Y)$

(2) The confidence of r is:

$confidence(r)=nsupport(X\cup Y)/nsupport(X)$

Given $minsup$ be the minimum support and $minconf$ be the minimum confidence, if $support(r)\geqslant minsup$ and $confidence(r)\geqslant minconf$, the rule r is considered to be an interesting rule (or pattern). For example, given $minsup$ =0.1 and $minconf$ =0.4, then in Table 1, $support(\{E\Rightarrow F\})$=0.109 8>0.1 and $confidence(E\Rightarrow F)$=0.45>0.4, so rule $E\Rightarrow F$ is an interesting rule.

4 MPSQAR Algorithm

Mining association rules usually includes two steps: (1) Find all the frequent item sets from data set; (2) Extract interesting rules from all the frequent item sets. Min-apriori algorithm is proposed for handling quantitative association rules directly[15], Min-apriori works as apriori[1]. Considering the weighted support of item set X defined in Definition 4, the apriori property does not make sense again.

Example 7 Given minimum support $minsup$=0.25, consider Table 4. Note that although $\{C,F\}\subset\{A,C,F\}$, $wsupport(\{A,C,F\})$=0.258 and $wsupport(\{C,F\})$=0.244, $\{A,C,F\}$ is a large item set while $\{C,F\}$ is not. In [1], $MINWAL(O)$ and $MINWAL(W)$ are proposed to tackle the weighted association rules with binary attributes. And the weight of each attribute is user specified while the weight for each attribute is produced by its distribution in this paper. Thus MPSQAR algorithm is proposed by revising the $MINWAL(O)$ for the weighted quantitative association rules in this paper. Similar to [16], let X, Y be item set, $minsup$ be the minimum support, and

$$nsupport(X)=\sum_{i=1}^{|T|}\min\{T_n(i,j)|j\in X\}$$

$$w(X)=\frac{1}{|X|}\sum_{j\in X} weight(j)$$

then $wsupport(X)=w(X)\times nsupport(X)$.

Let v be the maximum possible weight for any item set contains X, MPW is short for maximum possible weight, and $MPW(X)$ is used to denote v of X, then define $MPW(X)$ in mathematic form as $MPW(X)=\max\{w(Y)|X\subseteq Y\}$ Herein, it is easy to draw that $nsupport(X)\geqslant nsupport(Y)$ when $X\subseteq Y$. Also, we can infer

the lemma in the following.

Lemma 5 If $nsupport(X)<minsup/MPW(X)$, then X is not the true subset of any item set.

Proof let Y be any item set containing X, then $w(Y)\leqslant MPW(X)$. Since $nsupport(X)<minsup/MPW(X)$ and $nsupport(X)\geqslant nsupport(Y)$, so $nsupport(Y)<minsup/MPW(X)$. And because of $MPW(X)\geqslant w(Y)$, $nsupport(Y)<minsup/w(Y)$. So $nsupport(Y)\times w(Y)<minsup$, that is $wsupport(Y)<minsup$. As a result, Y is not a large item set. Especially, according to Lemma 3, since $weight(X)\leqslant 2$, so $w(X)\leqslant 2$ and $MPW(X)\leqslant 2$. As a result, if $nsupport(X)\leqslant minsup/2$, X cannot be the subset of any large item set. □

Similar to Apriori[1], MPSQAR employs large candidate $k-1$ item sets to produce candidate large k item sets. Let T be the data set, and T_n be the data set normalized from T, *Weights* be set of item weights, C_i be the candidate large i-item sets and L_i be the large i-item sets. Based on the above results, the *MPSQAR* algorithm is described as follows:

Algorithm MPSQAR (mining preserving semantic quantitative association rule)

Input: (1) T: the data set; (2) $minsup$: the minimum support

Output: a list of large item set L

Begin

$T_n=normalize(T)$;

$Weights[]=calculateWeight(T)$

$C_1=$singleItem$(T_n,minsup)$;

$L_1=$check$(C_1,minsup)$;

For (i=1;$|C_i|$>0;i++)

Begin

$C_{i+1}=join(C_i)$;

$C_{i+1}=prune(C_{i+1},minsup)$;

$L_{i+1}=check(C_{i+1},minsup)$;

$L=L\cup L_i$;

End

Return L

End

All the methods in MPSQAR are listed in the following:

(1) $normalize(T)$: use Equation (6) to normalize each value in T.

(2) $calculateWeight(T)$: according to Equation (8), get all the weights of all the attributes.

(3) $singleItem(T_n,minsup)$: based on all the single item set, following Lemma 5, the single item set X will be pruned if $nsupport(X)\leqslant minsup/2$ or $nsupport(X)\leqslant minsup/MPW(X)$.

(4) $prune(C_i+1,minsup)$: from candidate large (i+1)-item set, remove the item set X in following situations: ① existing a i-item set which is a subset of X does not occur in C_i. ② $nsupport(X)\leqslant minsup/2$. ③ $nsupport(X)\leqslant minsup/MPW(X)$.

(5) $join(C_i)$: similar to [1,3], return (i+1)-item sets.

(6) $check(C_{i+1},minsup)$: according to Equation (3), check data set T and the item set X which $wsupport(X)\leqslant minsup$ will be removed, return the large (i+1)-item sets.

5 Performance Study

Now we report the experimental results on the MPSQAR algorithm. It is implemented in Java. All the experiments are performed on HP Compaq 6510b with Intel(R) Core(TM)^2 Duo CPU 1.8 GHz and 1 G Memory and Windows Vista and run on both synthetic and real data sets.

(1) For synthetic data set, the values of each attribute will be 0 with a probability generated randomly ranging from 0 to 1. And the nonzero values of the attribute occur according to normal distribution whose mean and deviation are produced randomly. The range

of nonzero values, the number of transactions and number of attributes are all user-specified.

(2) For the real data set, we use the text data set called 19MclassTextWc which can be downloaded from WEKA data set page. In the data set, all the word count feature vectors have already extracted. So we can mine the patterns of the words occurrence.

To discuss the performance of experiment conveniently, some notations are given: *BI*, convert data set into binary data set depending on whether the value is greater than 0 firstly, then mine it with the apriori algorithm. *MA*: mine data set with min-apriori algorithm[15]. *QM*: normalize data set with Equation (6) and mine the data set without considering the weight of attribute. *WQ*: mine the data set employing MPSQAR algorithm.

Step 1 With data generator, 10 synthetic data sets containing 10 k transactions and 10 attributes are generated. And the 10 data sets vary with the number of quantitative attributes in each data set. Especially, when the number of quantitative attributes is 0, the data set can be viewed as a binary data set and when the number is 10, all the attributes are quantitative. Given *minsup* =0.3 and *minsup* =0.4, Variation in the number of large item sets on the synthetic data sets with changing number of quantitative attributes are shown in Fig.1 and Fig.2 respectively. As we can see, when the number of quantitative attributes is 0, *BI*, *QM* and *WQ* produce the same number of large item sets and that is in agreement with the Lemma 4, and the number for *MA* is greater than others due to its normalization way. For *BI*, there is no difference among different numbers of quantitative attributes, so *BI* cannot reflect the difference of quantitative attribute. Also, the number of large item sets for *WQ* is always greater than the one for *QM* due to the weight of attribute.

Step 2 Given the synthetic data set containing all 10 quantitative attributes and the real data set containing 50 quantitative attributes extracting from the real text data, then the variation in the number of large item sets with different *minsups* is shown in Fig.3 and Fig.4 respectively. As both figures shown, when the *minsup* increases, the number of the large item set de-

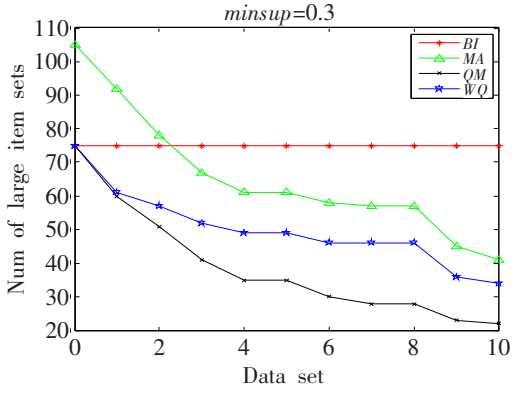

Fig.1 When *minsup*=0.3, the relationship between the number of large item sets and the number the of quantitative attributes

图 1 当 *minsup*=0.3 时,频繁项集数量和量化属性的数量关系

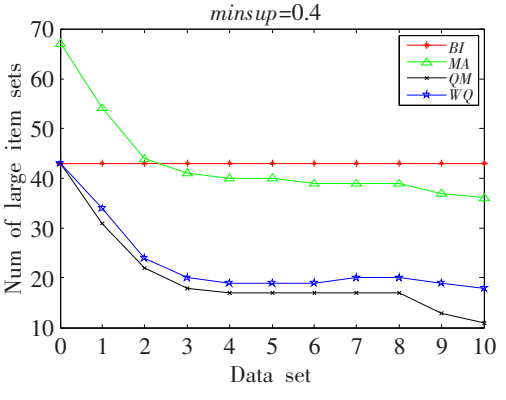

Fig.2 When *minsup*=0.4, the relationship between the number of large item sets and the number of the quantitative attributes

图 2 当 *minsup*=0.4 时, 频繁项集数量和量化属性的数量关系

creases. If the *minsup* gets close to 1, the number of large item sets for *BI*, *MA* and *WQ* approaches 0. However, the number for *MA* stops decreasing due to its side effect.

Step 3 With the data generator, 7 data sets containing 50 attributes and varying with different numbers of transactions from 100 k to 700 k. And execution time on these data sets is shown in Fig.5. Also, 9 data sets contain 100 k transactions and varying with changing number of attributes from 10 to 50. And execution time on these data sets is shown in Fig.6. From both figures, it shows that the new MPSQAR scales approximately linearly.

6 Conclusion and Future Work

Most existing work for quantitative association rules mining partition quantitative values into different bins and employ binary mining algorithm to extract the association rules. And the result rules just reflect the association relationship among these bins of different items rather than the association among different items due to the semantics loss of the partition of quantitative val-

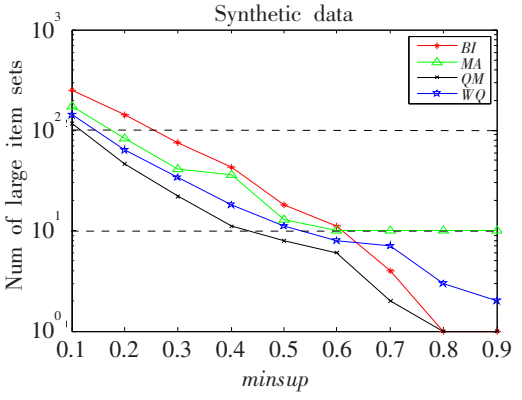

Fig.3 The relationship between the number of large item sets and the *minsup* over the synthetic data set

图 3 合成数据集上,频繁项集数量和支持度阈值的关系

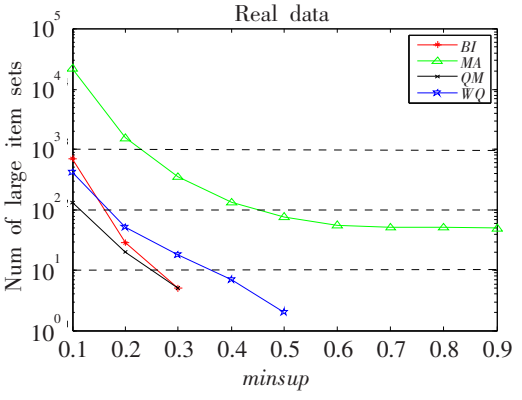

Fig.4 The relationship between the number of item sets and *minsup* over the real data set

图 4 真实数据集上,频繁项集数量和支持度阈值的关系

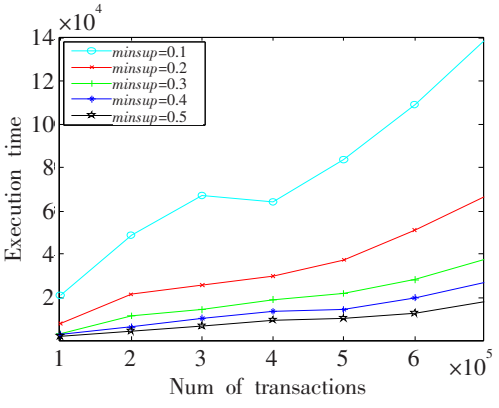

Fig.5 The relationship between the execution time and the number of the transactions

图 5 数据集大小与时间效率的关系

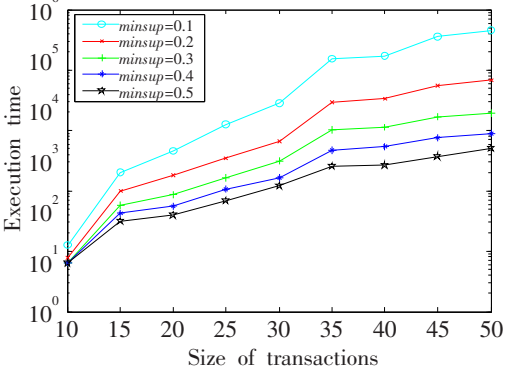

Fig.6 The relationship between the execution time and the size of the transactions

图 6 属性数量与执行时间效率的关系

ues. To deal with this problem, we have proposed the MPSQAR algorithm to mine the quantitative association rules directly by normalizing the quantitative values. MPSQAR also introduces a weight for each attribute according to the distribution of the attribute value and tackles the binary data and quantitative data uniformly without the side effect existing in Min-apriori. The experimental results show the efficiency and scalability of proposed algorithm. MPSQAR works well to extract the association among different attributes items rather than the partitions of items. However, the method of modeling the weight to reflect the distribution of attribute values is sensitive to the noise of attribute value. In the future, it is worthwhile to propose a better method to model the weight to be incorporated.

References:

[1] Agrawal R, Imieliski T, Swami A. Mining association rules between sets of items in large databases[C]//Proceedings of ACM SIGMOD, 1993:207-216.

[2] Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases[C]//Proceedings of the 20th VLDB Conference, 1994:487-499.

[3] Srikant R, Agrawal R. Mining generalized association rules[C]//Proceedings of the 21th VLDB Conference, 1995:407-419.

[4] Han Jiawei, Pei Jian, Yin Yiwen. Mining frequent patterns without candidate generation[C]//ACM SIGMOD, 2000:1-12.

[5] Han Jiawei, Kamber M. Data mining concepts and techniques[M]. 2nd ed. Beijing: China Machine Press, 2007.

[6] Miller R J, Yang Y. Association rules over interval data [C]//Proceedings of ACM-SIGMOD Int Conf Management of Data, 1997: 452-461.

[7] Srikant R, Agrawal R. Mining quantitative association rules in large relational tables[C]//Proceedings of SIGMOD 96, 1996:1-12.

[8] Kuok C M, Fu A, Wong M H. Mining fuzzy association rules in database[C]//Proceedings of ACM SIGMOD, 1999:41-46.

[9] Han Jiawei, Fu Yongjian. Discovery of mutiple-level association rules from large database[C]//Proceedings of the 21st VLDB Conference, 1995:420-431.

[10] Ke Yiping, Cheng J, Ng W. MIC framework: An information-theoretic approach to quantitative association rule mining[C]//Proceedings of ICDE'06, 2006.

[11] Aumann Y, Lindell Y. A statistic theory for quantitative association rules[J]. Journal of Intelligent Information Systems (JIIS), 2003: 225-283.

[12] Ruckert U, Richater L, Kramer S. Quantitative association rules based on half-spaces: An optimization approach[C]//Proceedings of ICDM'04, 2004:507-510.

[13] Wang Lian. An efficient algorithm for finding dense regions for mining quantitative association rules[J]. Computers and Mathematics with Application, 2005:471-490.

[14] Born S, Schmidt-Thieme L. Optimal discretization of quantitative attributes for association rules[C]//Proceedings of the Meeting of the International Federation of Classification Societies (IFCS), 2004:287-296.

[15] Han E H, Karypis G, Kumar V. Min-apriori: An algorithm for finding association rules in data with continuous attributes[R]. Department of Computer Science, University of Minnesota, Minneapolis, MN, 1997.

[16] Cai C H, Fu W C, Cheng C H. Mining association rules with weighted items[C]//Proc of IEEE Int'l Database Engineering and Applications Symposium, Cardiff, 1998:68-77.

ZENG Chunqiu was born in 1983. He received his B.S. degree in Computer Science and Technology from Sichuan University in 2006. He is a M.S. candidate in Computer Science & Technology at Sichuan University. His research interests include database, data mining.

曾春秋(1983-),男,四川资阳人,2006 年于四川大学获计算机科学与技术专业学士学位,目前为四川大学计算机学院应用专业硕士研究生,主要研究领域为数据库,数据挖掘。

TANG Changjie was born in 1946. He received his M.S. degree in Mathematics from Sichuan University in 1982. He is a full professor and doctoral supervisor at Sichuan University, Vice Director of Database Society of Chinese Computer Federation. His research interests include database, data mining, evolutionary computing.

唐常杰(1946-),男,重庆人,1982 年于四川大学数学系获理学硕士学位,四川大学计算机学院教授, 博士生导师,中国计算机学会数据库专委会副主任,四川省学术和技术带头人, 主要研究领域为数据库,数据挖掘,进化计算。

LI Chuan was born in 1977. He received his Ph.D. degree in Computer Science from Sichuan University in 2006. He is a lecturer at Sichuan University. His research interests include association, data warehousing, OLAPing and graph mining, etc.

李川(1977-),男,河南郑州人,2006 年 6 月获四川大学计算机应用专业博士学位,目前在四川大学计算机学院任讲师,主要研究领域为关联规则,数据仓库,在线分析处理,图挖掘等。

DUAN Lei was born in 1981. He received his Ph.D. degree in Computer Science from Sichuan University in 2008 His research interests include data mining, database, evolutionary computing, etc.

段磊(1981-),男,四川成都人,2008 年于四川大学获博士学位,主要研究领域为数据挖掘,数据库,进化计算等。