

An Effective Mining Algorithm for Weighted Association Rules in Communication Networks

Wu Jian

School of Communication and Information Engineering, University of Electronic Science and Technology of China
Chengdu, 610054, China
okmrj@hotmail.com

Li Xing ming

School of Communication and Information Engineering, University of Electronic Science and Technology of China
Chengdu, 610054, China
xingmingl@uestc.edu.cn

Abstract-The mining of weighted association rules is one of the primary methods used in communication alarm correlation analysis. With large communication alarm database, the traditional methods often treated each item evenly that makes the process of mining association rules time consuming. To improve the efficiency, items appearing in transactions are weighted using the analytic hierarchy process to reflect the importance of them which is more meaningful in some application. This paper implements a fast and stable algorithm to mining weighted association rules based on ISS(item sequence sets). Experiments on performance study will prove the superiority of the new algorithm.

Index Terms-Association rules, Data mining, Mining methods and algorithms, Rule-based databases

I INTRODUCTION

A. Background

Data mining is a process for discovering previously unknown and potentially useful abstractions from the content of large databases. Mining alarm association rules in networks can find out effective alarm rules which reflect the faults in networks rapidly. It is also validated that the system which provides the user the compact alarm rules with complete information is valuable and useful to the alarm correlation analysis and fault diagnosis, localization and recovery in networks. In modern networks, some of the alarms' attributes that have different levels and QoS requests result in different treatment of alarms. Therefore, traditional methods treated each item uniformly are no longer proper. Due to the problem, we put forward a solution by assigning corresponding weight to attribute of different importance called weighted association rule mining.

B. Association Rules

The basic task in mining for association rules is to determine the correlation between items belonging to a transactional database. Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of literals, called items. Let D be a set of transactions where each transaction T is a set of items such that $\forall T \in D, T \subseteq I$. An association rule is an implication of the form

$X \Rightarrow Y$, where $X \subseteq I, Y \subseteq I$ and $X \cap Y = \emptyset$. In general, every association rule must satisfy two user specified constraints, one is support and the other is confidence. The support of a rule $X \Rightarrow Y$ is defined as the fraction of transactions that contain $X \cup Y$, while the confidence is defined as the ratio $\text{support}(X \cup Y) / \text{support}(X)$. So, the target is to find all association rules that satisfy user specified minimum support and confidence values. We also call them *strong* rules to distinguish from the *weak* ones.

It has been shown that the problem of discovering association rules can be reduced to two sub-steps:

- ① Find all frequent itemsets for a predetermined support
- ② Generate the association rules from the frequent itemsets

The rest of this paper is organized as follows. Section 2 summarizes the related work in weighted ARM. Section 3 introduces the AHP to determine the weights of all items. Section 4 presents the proposed weighted association rule mining method ISS. Experimental and comparison results are reported and discussed in Section 5. The last section concludes this paper.

II RELATED WORK

When we compute the weighted support of the rule, we can consider both the support and the weights factors. The classical Apriori algorithm^[1] for finding binary association rules depends on the downward closure property which governs that subsets of a frequent itemset are also frequent. However, it is not true for the weighted case in our definition, and the Apriori algorithm cannot be applied. Therefore, another algorithm MINWAL^[2] is designed for this purpose.

Given a set of items $I = \{i_1, i_2, \dots, i_m\}$, we assign a weight w_j for each item i_j , with $0 \leq w_j \leq 1$, where $j = \{1, 2, \dots, m\}$, to show the importance of it. We define the weighted support for the weighted association rule $X \Rightarrow Y$ is

$$\left(\sum_{i_j \in (X \cup Y)} \omega_j \right) \bullet (\text{Support}(X \cup Y)) \quad (1)$$

A k-itemset X is called a frequent itemset if the weighted support of such itemset is no less than the minimum weighted support (ω minsup) threshold, or

$$\left(\sum_{i_j \in X} \omega_j \right) \bullet (\text{Support}(X)) \geq \omega \text{ min sup} \quad (2)$$

An association rule $X \Rightarrow Y$ is called an interesting rule if $X \cup Y$ is a frequent itemset and the confidence of the rule is greater than or equal to a minimum confidence threshold.

In this situation, we want to get a balance between the two measures: weight and support. We have three parameters to consider in the weighted association rule: weights of items, support of itemsets and the confidence factor. Given a database with T transactions, we define the support count of a frequent k-itemset X to be the transaction number containing X, and it must satisfy:

$$\text{Support}(X) \geq \frac{\omega \text{ min sup} \times T}{\sum_{i_j \in X} \omega_j} \quad (3)$$

Let I be the set of all items. Suppose that Y is a q-itemset, where $q < k$. In the set of the remaining items (I-Y), let the items with the (k-q) greatest weights are $i_{r_1}, i_{r_2}, \dots, i_{r_{k-q}}$. We can say the maximum possible weight for any k-itemset containing Y is (4) in which the first sum is the sum of the weights for the q-itemset Y, and the second sum is the sum of the (k-q) maximum remaining weights.

$$W(Y, k) = \sum_{i_j \in Y} \omega_j + \sum_{j=1}^{k-q} \omega_{r_j} \quad (4)$$

From (3), the minimum count for a large k-itemset containing Y is given by

$$B(Y, k) = \left\lceil \frac{\omega \text{ min sup} \times T}{W(Y, k)} \right\rceil \quad (5)$$

We called this the k-support bound of Y. We take an upper bound of the value since the function B(Y, k) is an integer. The MINWAL algorithm for mining weighted association rules can be established according to the above properties of the k-support bound for all possible k-itemsets. Y is called a potentially frequent itemset if the support count of it satisfies:

$$\text{Support}(Y) \geq B_{\min}(Y, k) \quad (6)$$

$q \leq k \leq L$

It could easily be seen that MINWAL is also an iteration process like Apriori, the only difference is all candidate k-itemsets are generated by potentially frequent (k-1)-itemsets while Apriori does by frequent (k-1)-itemsets. Since MINWAL could not reduce the scan number of the database, we suggest the ISS algorithm to solve this problem in section 4.

III THE ANALYTIC HIERARCHY PROCESS—AHP

C. The Strategy of Setting Weights

As our research is based on the communication system, the equipment which occurs alarms in the network, the associations among the equipments and the level of alarms

are considered together to decide the weights of alarms. We adopt the method of AHP to set the weight of single alarm occurs in single equipment by associated attributes at different levels to realize both qualitative and quantitative analysis.

At present, the most widely used method to determine the subjective weight of item, is the AHP proposed in 1970s by the famous mathematician T.L.Saaty^[4]. This method has its unique superiority to determine various weight of items at different level in a big system, fast and simple, easy to calculate, therefore it has been widely used^[5]. But this method depends too much on the subjective aspect that this has become its greatest shortcomings. For example, when different person use the AHP method to get the weight of indexes in the same index system, the structure of the judgment matrix they built may varies from each other, thus the index weight would be different. The above situation could be avoided by changing the artificial judgment matrix method to the objective judgment matrix method.

When the AHP method is used to calculate the weight of the items, it is important to know both the degree of relative significance among different values of one attribute and the degree of relative significance among evaluation attributes. Thus the concept of degree of relative significance is proposed in this paper, namely the difference of two of the degree. Suppose the set of significance values of a certain condition attribute is C, C consists of n different values as c_1, c_2, \dots, c_n , then the degree of relative significance of these values could be expressed by the equation:

$$\Delta c_{i,j} = c_j - c_i \quad (i, j=1, 2, \dots, n) \quad (7)$$

Because the value scope of the degree of significance of attributes is normalized to [0,1], the value scope of the degree of relative significance is [-1,1].

D. Judgments Matrix Building

Suppose that the value c_i is more important than the value c_j ($i, j=1, 2, \dots, n$), then the degree of relative significance between c_i and c_j could be expressed as $\Delta c_{i,j} > 0$ and would increase with $\Delta c_{i,j}$, on the contrary when $\Delta c_{j,i} < 0$ and the degree of relative significance between c_j and c_i would decrease with $\Delta c_{j,i}$. Therefore, in the communication network, the judgment matrix of the AHP could be built according to the degree of relative significance of two values. If the degree of relative significance between c_i and c_j is expressed as $\Delta c_{i,j} > 0$, then suppose the degree of significance of value c_i is M times more than that of value c_j ($M=1, 2, \dots, 10$). In other word, the degree of significance of value c_j is 1/M of that of value c_i . The value scope of the degree of relative significance Δc and the corresponding M are listed in Table 1.

After comparing of the two values, the judgment matrix B could be obtained according to the value of Δc . Its form is showed as Table 2 where b_{ij} is the degree of relative significance between c_i and c_j which could take the

corresponding value M from Table 1.

To calculate the weight of every c_i through judgment matrix could be summed up as the following steps:

① Once the judgment matrix was built, calculate the product $M_i(i = 1, 2, \dots, n)$ of every row by the following formula:

$$M_i = \prod_{j=1}^n b_{ij} \tag{8}$$

② Then compute the Nth root of all M_i :

$$\overline{W}_i = \sqrt[n]{M_i} \tag{9}$$

③ At last, normalize vector $\overline{W} = [\overline{W}_1, \dots, \overline{W}_n]^T$ like:

$$W_i = \frac{\overline{W}_i}{\sum_{j=1}^n \overline{W}_j} \quad (i=1, 2, \dots, n) \tag{10}$$

Value W_i represents the corresponding weight of value c_i of attribute C.

Table1. The degree of relative significance ΔC and corresponding M

$\Delta\sigma$	M	$\Delta\sigma$	M
$0.0 \leq \Delta\sigma \leq 0.1$	1	$-0.1 \leq \Delta\sigma \leq 0.0$	1
$0.1 < \Delta\sigma \leq 0.2$	2	$-0.2 \leq \Delta\sigma < -0.1$	1/2
$0.2 < \Delta\sigma \leq 0.3$	3	$-0.3 \leq \Delta\sigma < -0.2$	1/3
$0.3 < \Delta\sigma \leq 0.4$	4	$-0.4 \leq \Delta\sigma < -0.3$	1/4
$0.4 < \Delta\sigma \leq 0.5$	5	$-0.5 \leq \Delta\sigma < -0.4$	1/5
$0.5 < \Delta\sigma \leq 0.6$	6	$-0.6 \leq \Delta\sigma < -0.5$	1/6
$0.6 < \Delta\sigma \leq 0.7$	7	$-0.7 \leq \Delta\sigma < -0.6$	1/7
$0.7 < \Delta\sigma \leq 0.8$	8	$-0.8 \leq \Delta\sigma < -0.7$	1/8
$0.8 < \Delta\sigma \leq 0.9$	9	$-0.9 \leq \Delta\sigma < -0.8$	1/9
$0.9 < \Delta\sigma \leq 1.0$	10	$-1.0 \leq \Delta\sigma < -0.9$	1/10

Table2. The form of the judgment matrix B

Index	c_1	c_2	c_3	c_4	c_5
c_1	1	3	5	7	9
c_2	1/3	1	3	5	7
c_3	1/5	1/3	1	3	5
c_4	1/7	1/5	1/3	1	3
c_5	1/9	1/7	1/5	1/3	1

Regarding the judgment matrix B, the characteristic root and the characteristic vector could be calculated out through:

$$BW = \lambda_{\max} W \tag{11}$$

In (11), λ_{\max} is the biggest characteristic root of the judgment matrix B, W is the characteristic vector, the component w_i in W is the weight of the attribute value c_i . The calculation method of the matrix characteristic root and the characteristic vector could be found in a general linear algebra book.

When the judgment matrix B is of complete uniformity, that is $\lambda_{\max} = n$. But it is impossible in the ordinary circumstances. In order to examine the uniformity of the judgment matrix, the uniform index should be calculated out as:

$$CI = \frac{\lambda_{\max} - n}{n - 1} \tag{12}$$

In (12), when $CI = 0$, the judgment matrix is of complete uniformity; otherwise the bigger CI is, the worse uniformity of the judgment matrix is. In order to check whether the uniformity of judgment matrix is satisfying, the comparison between CI and the average stochastic uniform index RI (Table 3) is necessary. The ratio of CI and RI at the same step is called the stochastic uniform proportion of judgment matrix, as CR. Generally, when $CR = CI/RI < 0.10$ the judgment matrix has the satisfying uniformity, and the weight W satisfies the objective, scientific and credible requirement. Otherwise, we have to modify the judgment matrix.

Table3. The average stochastic uniform index RI

Step	1	2	3	4	5
RI	0	0	0.58	0.89	1.12
Step	6	7	8	9	10
RI	1.24	1.32	1.41	1.45	1.49
Step	11	12	13	14	15
RI	1.52	1.54	1.56	1.58	1.59

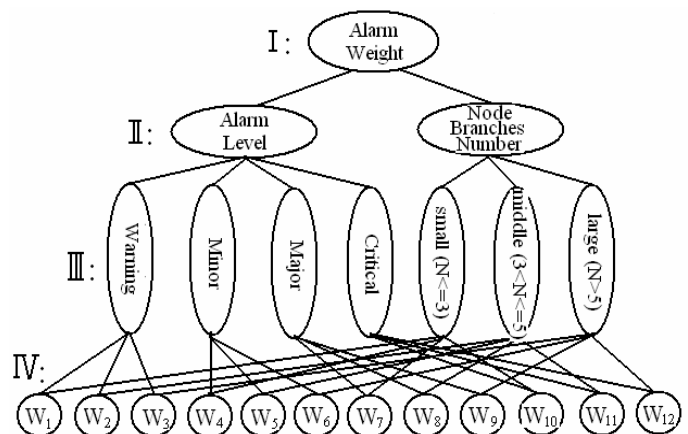


Figure 1 Hierarchy of alarm weights

E. Calculation The Weight of Items

The correct determination of weight should be the reflections of both the objective information of network and the experts' subjective judgment. A method of calculating the relative significance of items is proposed in this paper: the traditional analytic hierarchy process (AHP) method of subjective weight method is improved to enable the judgment matrix built to be objective and the weight of item to be more objective, scientific and credible.

Our purpose is to set weights for every alarm that contains several attributes, and we only consider attributes which can reflect the relative significance of certain alarm among others. Here we select two: one is alarm level, the other is number of branches of alarm node. Since there are different levels and numbers of alarms, the hierarchy of alarm weights is described in Fig.1.

The weights of items should be calculated according to the following principle.

- ① Get the degrees of relative significance among different values of alarm level and number of branches of alarm

node to build the corresponding judgment matrix B_1 and B_2

- ② Compute the weight of each value of those two attributes by the above method, then express them as $W_i = [W_{i1}, W_{i2}, W_{i3}, W_{i4}]$ and $W_n = [W_{n1}, W_{n2}, W_{n3}]$
- ③ Consider the degrees of relative significance among evaluation attributes. Here define them as S_1 and S_2 for alarm level and number of branches of alarm node, respectively.
- ④ Calculate the weights of all alarms as multiplying weights of the values of different alarm attributes by relative significance degrees of corresponding attributes and adding all of them together. For instance, the weight of alarm which level is W_{li} as well as number of node branches is W_{nj} is given by $=S_1 * W_{li} + S_2 * W_{nj}$.
- ⑤ Since $0 < \omega \text{ minsup} \leq 1$, we should normalize every weight of alarms. Suppose there are k attributes, and the n -th attribute has L_n different weights and relative significance degree S_n . The total value of all weights is $Sum = S_1 * L_2 * L_3 * \dots * L_k + S_2 * L_1 * L_3 * \dots * L_k + \dots + S_k * L_1 * L_2 * \dots * L_{k-1}$, and in our system the specific value should be $Sum = S_1 * L_2 + S_2 * L_1$. The normalize weights will be described as:

$$W_i = \frac{\overline{W}_i}{Sum} \quad (i=1,2, \dots, n) \tag{13}$$

To show the applicability of these weights, we compare three different ways to set weights of items and mine weighted frequent itemsets from them using MINWAL^[2]. Fig.2 shows our method is better for getting weighted frequent itemsets that have strong correlations than the two others. In way 1, the weights are simply decided by alarm levels. In way 2, the weights are decided by alarm levels and alarm types. However, the relative significances among different alarm types are not so obvious which makes for similar mining results between way 1 and way 2. As for way 3, the weights reflect information not only about the network topology, but also the degrees of relative significance of alarms. It has strong ability to distinguish the importance of alarms of different attribute-value-combinations, and could self adapt to the dynamic change of the network. Therefore, with the increase of $\omega \text{ minsup}$, the number of weighted frequent itemsets in way 3 drops slower than that in way 1 and way 2.

IV OUR NEW ALGORITHM—WARM

Dynamic counting technique has been explored by Brin et al. who developed the DIC algorithm^[3] for itemset counting in a single database. While Apriori requires the same number of passes through the database as the size of the maximum large itemsets, dynamic counting allows candidates of various sizes to be counted at the same time in each pass. This results in the reduced number of passes through the database. What DIC does is just to create possible new candidate itemsets at every M -transaction interval while the Apriori algorithm creates new candidate itemsets only at the end of a pass through the database.

By the inspiration of DIC, we proposed an algorithm for mining weighted association rules-WARM based on

ISS(item sequence sets).

WARM algorithm has the following inputs and output. In order to describe easily, we give the following definitions and Notations:

D	the database
ω	the set of item weights
L_k	set of frequent k -itemsets
Support(X)	number of transactions containing itemset X
$\omega \text{ minsup}$	weighted support threshold
minconf	confidence threshold

Inputs: A database D with the transactions T, two threshold values $\omega \text{ minsup}$ and minconf, weights of the items w_i

Output: A list of interesting rules.

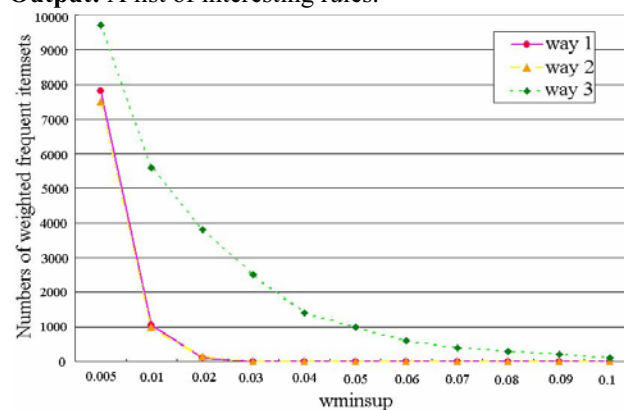


Figure 2 Numbers of weighted frequent itemsets with different ways of weight setting

There are two linear data structure ISS and FSS in our algorithm. The former is all the itemsets gotten from scan of the database and the latter is all the frequent itemsets generated from the algorithm. The basic idea of WARM is adding the item sequence sets of transaction t after scan it to ISS and counting the support counts of all its subsets to put those whose counts satisfy the $\omega \text{ minsup}$ into FSS. It just likes to set the M -interval in the above DIC algorithm to 1, and can calculate all the support counts of itemsets length from 1 to $|t|$ belong to transaction t . The procedure is described as follows.

WARM Algorithm(D, ω , $\omega \text{ minsup}$, minconf, L)

```

{
  ISS = ∅; FSS = ∅;
  for all IS ∈ D //get item sequence sets of transactions in D
  {
    add(IS,ISS);
    gen_fre(IS,ISS,FSS, ω minsup);
  }
  L = FSS;
  Rules(Support,L);
}

```

The sub-procedure **add** is used for scanning transactions in D one by one and adding them to ISS.

```

add(IS,ISS)
{
  sup_count (IS) = 1; flag = 0;
  for all IS1 ∈ ISS

```

```

{
  if(IS= =IS1)
  {
    sup_count (IS1) ++;
    flag=1;
  }
}
if(flag= =0) ISS=ISS U {IS};
}

```

The sub-procedure **gen_fre** makes use of IS to select frequent itemsets in ISS, then add them to frequent item sequence sets FSS. Along with the addition of frequent itemsets, FSS will become bigger and bigger. It works as follows.

```

gen_fre(IS,ISS,FSS,  $\omega$  minsup)
{
  for all IS* subset{IS}
  {
    if(IS*  $\notin$  subset{FSS})
    {
      sup_count (IS*)=0
      for all IS** ISS
      if (IS*  $\subseteq$  IS**)
        sup_count (IS*) += sup_count(IS**);
      if( (  $\sum_{i_j \in IS^*} \omega_j$  ) • (Support(IS*))  $\geq$   $\omega$  min sup )
        FSS=FSSU {IS*};
    }
  }
}

```

The sub-procedure **subset{IS}** is to get the aggregation of all subsets of IS, it operates as follows.

```

Set subset(IS)
{
  Set<String> subsetIS=new HashSet<String>();
  N=|IS|; //the length of itemsets in IS
  k=0;
  String[] S=new String[2N-1];
  for(i=0;i<S.length;i++)
    S[i]="";
  char[] c=IS.toCharArray();//construct array that makes up
  of all single items in IS
  for(i=1;i<2N;i++)
  {
    for(j=0;j<N;j++)
      if((i>>j)&1)
        S[k]+=c[j];
    subsetIS.add(S[k]);
    k++;
  }
  return subsetIS;
}

```

Rules (Support,L): Find the rules from the large itemsets L, similar to [1].

WARM only needs to scan the transaction database once, and need not to generate the candidate itemsets. Therefore, we can conclude that the excellence of our algorithm over the MINWAL algorithm is not only contributed by the use of dynamic counting technique, but also the whole design

of the algorithm.

V PERFORMANCE STUDY OF EDMA

F. Generation of Synthetic Data

In our research, we used synthetic databases to evaluate the performance of different algorithms. Fig.3 shows the topology of our network and the alarm data is generated as following:

- ◇ Alarm occurs in one of the blue nodes randomly as well as the alarm level and alarm type
- ◇ Alarms of the blue nodes cause corresponding alarms in same level of all neighboring nodes and will not be transmitted further

After generating alarms, it comes to the problem of alarm's synchronization which was settled by setting time window and slip length. The attributes of the alarm that reflect the faults were picked out to form an item of an alarm transaction and the redundant alarms were got rid of by alarm compressing. The alarm's pretreatment was carried out to transform the alarm database into alarm transaction database which make ready the data for weighted alarm association rules mining.

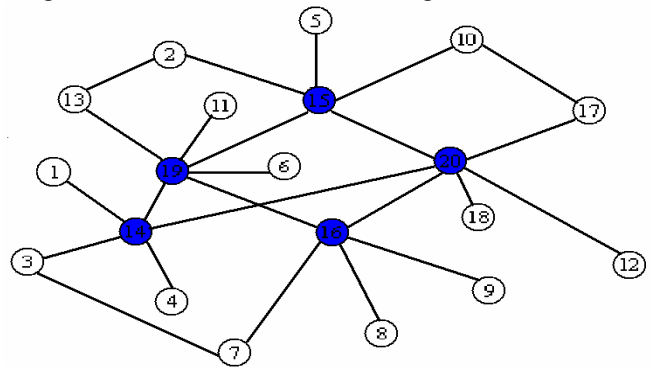


Figure 3 Topology of our experimental network

G. Experimental Results

We have extensively studied our algorithm's performance by comparing it with the FP-tree algorithm^[6] as well as MINWAL^[2] and consider the superiority of it. FP-tree only scans the database twice and never generates the huge candidate itemsets, therefore reduces the temporary occupation of memory space as well as I/O cost needed for mining. However, the memory space can not be downgraded sometimes and the tree is a nonlinear structure that needs link list to storage, which increases the overhead of storing the link information.

In order to control different parameters in the experimental setup, we use synthetic databases and weights formed above in the experiment. We also use the same item weights is to keep this factor constant to confirm the efficiency of WARM with different transaction databases.

We study the effect of different values of ω minsup, number of transactions and items, etc., on the processing time, for algorithms weighted FP-tree, MINWAL and WARM.

We performed some empirical tests on above synthetic data in order to evaluate the effectiveness and applicability of the proposed approach and to analyze its scalability. All the experiments were conducted on a Pentium IV 2GHz CPU with 512MB of memory and running Windows XP

and the algorithm was written in Java.

First of all, our algorithm is based on the theory and operation of item sequence sets (ISS). We generate transaction database ranging from 10K to 100K and set ω minsup=0.2, then examine how the method scales up and compare the performance of them. Fig.4 indicates that not only the whole execution time of ISS is better than Apriori, but also with increase of transactions the increasing scope of execution time of ISS is less than that of Apriori. As for weighted case, the result is a little different. The more item sequences are, the better WARM performs than MINWAL. Because when transactions grow, there are more weighted potentially frequent itemsets for the latter to generate and calculate. We can also see that WARM is more scalable than MINWAL.

Then Fig.5 describes that, with the increase of transactions the increment of FSS is mild which is much less than its theoretical value. Because our database is generated randomly, it almost has no repetition. If the repetition of database is large, it will surely cause the decline of length of ISS. Therefore, how to control the space that ISS occupy is the most needed improvement for our algorithm.

In the following, we examine how the performance varies with respect to different number of items and ω minsup while all other things being equal. With transaction database is 100K for all cases, a series of items range from 100 to 1K. The execution time increases with the number of items since the more the items, the larger the candidate itemsets, the global FP-tree as well as the item sequence. As all of them grow directly with the items, the execution time in the count searching would be increased, which is shown in Fig.6. From this figure, we can see the execution time of WARM is about 78% to 93% and 33% to 73% faster than MINWAL and weighted FP-tree, respectively. This denotes WARM has a steady advantage over MINWAL and weighted FP-tree in terms of different number of items and the execution time is much less than its theoretical upper limit. The slope ratio for each different size is almost similar. As the execution time is given in ln(sec), and it increases with the number of items linearly with ln scale, implying that the complexity of the algorithms is exponential in the number of items.

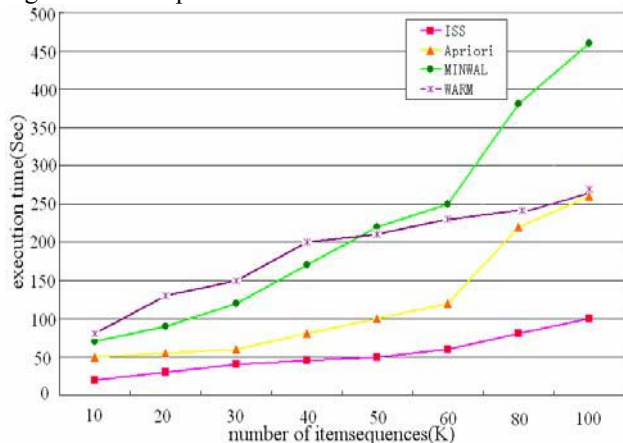


Figure 4 Runtime with itemsequences scale-up

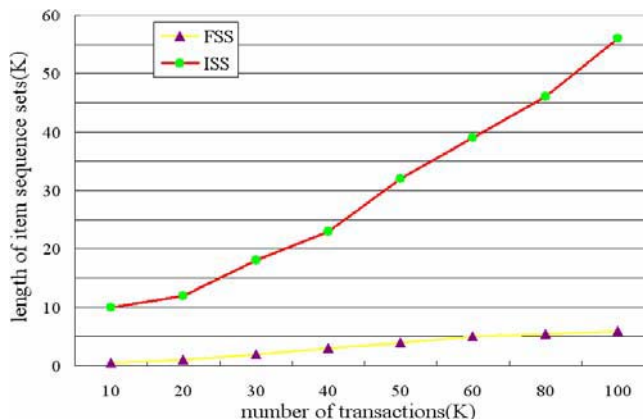


Figure 5 Space of ISS and FSS versus transactions

Fig.7 displays the decreasing trend of the execution time of MINWAL when the support threshold increases. As the threshold increases, the candidate itemsets decrease. Then the execution time would decrease because of the decreasing searching time. However, weighted FP-tree and WARM are based on the conditional pattern-base (not the conditional FP-tree) and item sequence sets respectively, which are not affected by ω minsup. Therefore, the whole execution times of them almost keep the same among various ω minsup thresholds. The reduction of time about 1% to 49% and 5% to 18% are witnessed in WARM compare to MINWAL and weighted FP-tree, respectively.

The next experiment deals with relationship between the number of frequent itemsets involving some given items and these items' weight is plotted in Fig.8, when ω minsup=0.2 and transaction database is fixed at 10K. Here we let the respective weight of the five items (alarm1, alarm2, alarm3, alarm4 and alarm5) vary, while weights of other items held equally. Figure 8 shows the number of frequent itemsets with regard to specific item grows with the increase of the item's weight, which is as we expected. Each frequent itemset is a potential association rule, so the bigger the weight of an item is, the more the relative association rules are under predefined minconf. This also means users' subjective preference can be added into the procedure of data mining by adjusting weights.

At last, Fig.9 compares the number of weighted frequent itemsets for different ω minsup. We can see that the shorter the length of frequent itemsets is, the smoother the corresponding curve is. It is natural because higher requirements are asked for the candidate itemsets that include more items, which makes these itemsets are more sensitive to the change of the ω minsup.

VI CONCLUSION

In the alarm correlation in communication networks, the weight of alarms may affect the accuracy and the validity of the results directly. Moreover, the proper determination of weight should be the reflections of both the objective information of alarm and the experts' subjective judgment. The proposed weighted association rule mining algorithm -WARM enable the judgment matrix built to be objective and the weight of alarm to be more flexible, natural and understandable. It is easy to add users' guide to data mining procedure, so it is quicker to find association rules interested by users, and reduce the work burden of

choosing interesting rules. In addition, for each support value is normal size, similar to that in apriori algorithm, the ω minsup is also easy to provide or predict by the users or experts in the proposed method. Experimental results have demonstrated it outperforms the other two algorithms.

Beside, as in our communication network different data sets may spread over various geographical sites and be periodically and continually updated. Distributed mining as well as dynamic mining of association rules can be further considered.

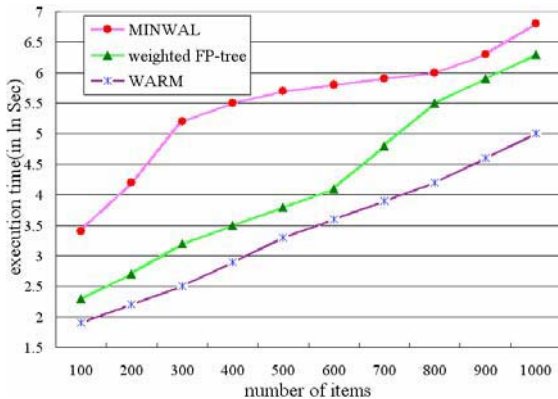


Figure 6 Overall execution time with number of items scale-up

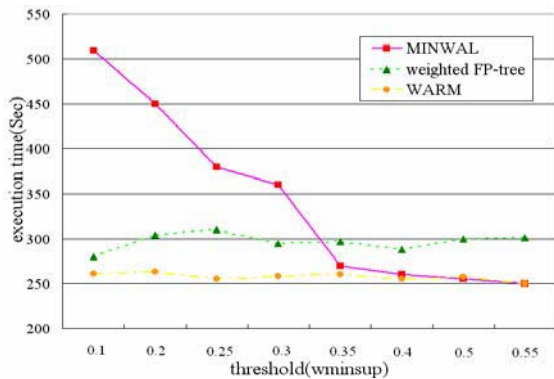


Figure 7 Overall execution time with ω minsup scale-up

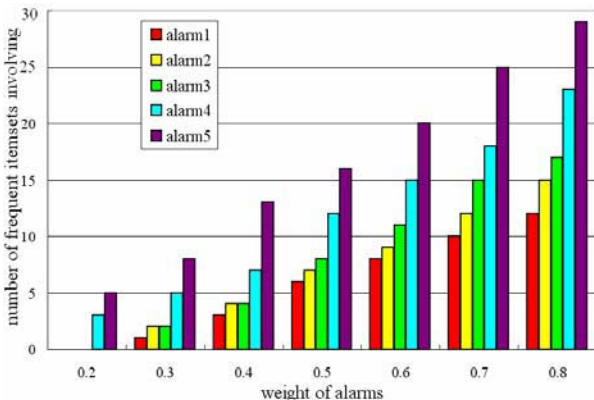


Figure 8 Number of frequent itemsets versus weight

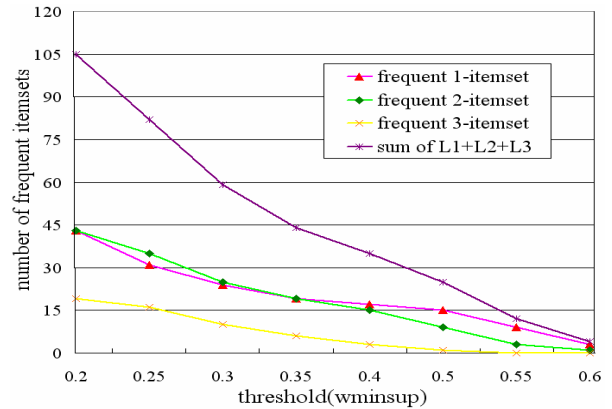


Figure 9 Number of frequent itemsets with ω minsup scale-up

ACKNOWLEDGMENTS

This work was supported by the Republic of China National Science Foundation under Contract No.60572091.

REFERENCES

- [1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proceedings of the 20th VLDB Conference, pages 487-499, 1994
- [2] C.H.Cai, Ada W.C. Fu, C.H. Cheng and W.W. Kwong, Mining Association Rules with Weighted Items. Proceedings of the 1998 International Symposium on Database Engineering & Applications, Cardiff, Wales, 1998, pp. 68-77.
- [3] S.Rrin, R.Motwani, J.D.Ullman, and S.Tsur. Dynamic itemset counting and implication rules for market basket data. In Proceedings of ACM SIGMOD International Conference on Management of Data, pages 255-264, AZ, 1997
- [4] Saaty S, Thomas L. The Analytic Hierarchy Process [M]. New York:McGraw-Hill Company, 1980.
- [5] S.L.Tung, S.L.Tang, A comparison of the Saaty's AHP and modified AHP for right and left eigenvector inconsistency. European Journal of Operational Research, ELSEVIER, 1998,123-128.
- [6] J.Han, J.Pei, Y.Yin, and R.Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach [J]. Data Mining and Knowledge Discovery, 2003.8(1):53-87
- [7] Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Mining optimized association rules for numeric attributes. Technical Report 1623-14, IBM Tokyo Research Laboratory, 1996.
- [8] J. Han, M. Kamber, and J. Chiang. Mining multidimensional association rules using data cubes. Technical report, Database Systems Research Laboratory, School of Science, Simon Fraser University, 1997.
- [9] J.S. Park, M-S. Chen, and P.S. Yu. An effective hash-based algorithm for mining association rules. In Proceedings of ACM SIGMOD, pages 175-186, 1995.
- [10] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In Proceedings of the 21th International Conference on Very Large Data Bases, pages 432-444, 1995.
- [11] W.J. Frawley, G. Piatetsky-Shapiro and C.J.Matheus, Knowledge discovery in databases: an overview, The AAAI Workshop on Knowledge Discovery in Databases, 1991, pp.1-27.
- [12] A. Gyenesei, Mining weighted association rules for fuzzy quantitative items, Proceedings of PKDD Conference, September 13-16, 2000, Lyon, France, 2000, pp.416-423.

- [13] R. Srikant and R. Agrawal, Mining quantitative association rules in large relational tables, The 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Canada, June 1996, pp. 1-12.
- [14] S.-F. Lu, H. Hu and F. Li, Mining weighted association rules, *Intelligent Data Analysis*, 5, 2001, pp. 211-225.
- [15] T.-P. Hong and S.-C. Chi. Trade-off between computation time and numbers of rules for fuzzy mining from quantitative data. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems* 9, 2001 587-604.
- [16] Mannila H., Toivonen H., and Verkamo A.I. Discovering frequent episodes in sequences. In *Proceeding of the First International Conference on Knowledge Discovery and Data Mining*, pages 210-215, Montreal, Quebec, 1995.
- [17] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceeding of the Fifth International Conference on Extending Database Technology*, Avignon, France, 1996
- National Science Foundation under Contract No.60572091 in 2006. Her main job included: Study the mining of the alarm weighted association rules Research the association rules from distributed data sources Learn the association rules from self-adaptation for dynamic network resource. Following are selected recent publications:
- Wu Jian and Tu Xiao dong. Analysis and Realization to Security of Radio Frequency Identification system. In *Proceedings of the 6th International Conference on ITS Telecommunications*, pages 222-225, China, 2006
 - Wu Jian and Li Xing ming. A Novel Algorithm for dynamic Mining of Association Rules. In *Proceedings of the 1st International Workshop on Knowledge Discovery and Data Mining*, pages 94-99, Australia, 2008
 - Wu Jian and Li Xing ming. An Efficient Association Rule Mining Algorithm In Distributed Databases. In *Proceedings of the 1st International Workshop on Knowledge Discovery and Data Mining*, pages 108-113, Australia, 2008

Wu-Jian (female) was born in Chengdu , China on June 20, 1981 . She received the B.S. degree in information engineering from Sichuan University in 2004. She is currently a Ph.D. candidate of Key Laboratory of Broadband Optical Fiber Transmission and Communication Networks of UESTC, specializing in Data Mining & Knowledge Discovery.

She worked in the project “The alarm correlation in communication networks based on data mining” supported by

Li-Xing-ming (male), received the M.S. and Ph.D. degrees in school of automation engineering from Tsinghua University in 1993 and in telecom engineering from Beijing University of posts and telecommunications in 1996, respectively. He is currently a professor and Doctor’s Tutor of University of Electronic Science and Technology of China. He is also the senior member of China Institute of Communications. His research interests include Current Communication Network, Optical Transmission Network, Automatic Switched Optical Network