# Transaction-item Association Matrix-Based Frequent Pattern Network Mining Algorithm in Large-scale Transaction Database

WEI-QING SUN[1], CHENG-MIN WANG[1], TIE-YAN ZHANG[2], YAN ZHANG[1]
[1] Department of Electronic, Information & Electrical Engineering
Shanghai Jiaotong University
800 Dongchuan Road, Shanghai
CHINA
neilswq@gmail.com, wangchengmin@sjtu.edu.cn, zhang_yan@sjtu.edu.cn
[2] Shenyang Institute of Engineering
18 Zhengyi San Road, Shenyang, Liaoning
CHINA
tieyanzhang@163.com

*Abstract:* - To increase the efficiency of data mining is the emphasis in this field at present. Through the establishment of transaction-item association matrix, this paper changes the process of association rule mining to elementary matrix operation, which makes the process of data mining clear and simple. Compared with algorithms like Apriori, this method avoids the demerit of traversing the database repetitiously, and increases the efficiency of association rule mining obviously in the use of sparse storage technique for large-scale matrix. To incremental type of transaction matrix, it can also make the maintainment of association rule more convenient in the use of partitioning calculation technique of matrix. On the other and, aiming at the demerits in FP-growth algorithm, this paper proposes a FP-network model which compresses the data needed in association rule mining in a FP-network. Compared with the primary FP-tree model, the FP-network proposed is undirected, which enlarge the scale of transaction storage; furthermore, the FP-network is stored through the definition of transaction-item association matrix, it is convenient to make association rule mining on the basic of defining node capability. Experiment results show that the FP-network mining association rule algorithm proposed by this paper not only inherits the merits of FP-growth algorithm, but also maintains and updates data conveniently. It improves the efficiency of association rule mining significantly.

*Key-Words:* - association rule, association matrix, data mining, FP-growth algorithm, FP-network algorithm, frequent itemset

## 1 Introduction

With continuous development of database technology and widely application of database management systems, the amount of data stored in databases increases sharply. But faced with such massive data, tools which can make analysis and treatment to them is few [1]. Due to the limitation of tools used at present, people can not mine many important information concealed in huge amount of data, although they are very valuable for people's decision.

To solve these problems, Knowledge Discovery in Database (KDD) was developed. KDD is also called Date Mining (DM). There are small differences between KDD and DM, but usually they are used without distinguish [2].

The purpose of data mining is to find out the implicit, unknown and valuable knowledge and rules. These rules contain the special relationship among itemsets in database, reveal some important information and provide evidence for management decision, market planning and financial prediction, etc. The existing data mining methods are mainly based on association rule [3], neural network [4-6], decision tree [7-9], decision tree-neural network [10-13], Rough set [14-16] and others [17-21].

Association rule is an important research subject put forward by R. Agrawal in [3] in 1993, which can find out the close relationship among itemsets in databases. Plenty of research works on mining association rules are made by researchers in the field of data mining from then on [22-28]. The key research problem in association rule algorithm is to generate all the frequent itemsets.

The existing association rule mining methods are mainly AIS [3], SETM [25] and others [26, 27]. The most famous one is the Apriori algorithm [28-32] put forward by Agrawal. The main demerit of

Apriori algorithm is to find huge amount of candidate generations. When the database is large, there exists the problem of combination explosion. And the need of searching the database for many times also increases the difficulty of calculation. Aiming at such demerits, many improved methods are proposed [33-35].

Therefore, J. Han put forward a method of creating frequent pattern sets through frequent pattern tree [36, 37]. FP-growth algorithm compresses the database which provides frequent items to a FP-tree, then begins for the initial postfix pattern to establish condition pattern groups, and then forms condition FP-tree, mining on the tree recursively. Its main merits are:

1) It does not need to produce candidates, only to construct FP-Tree and condition FP-Tree, and produce frequent pattern by visiting FP-Tree recursively;

2) Only two traversing to the whole transaction database is enough, one time to produce frequent1-itemset and the other time to create FP-Tree, consequently decrease the visit times to the database apparently.

Although FP-growth algorithm does not produce candidate generations and traverses the database only two times, its demerits are mainly incarnated on the difficulties of updating and maintaining the FP-Tree; besides, the process of forming the tree and association rule mining is comparatively complex.

Data mining technology splits the difference between calculation efficiency and accuracy. Method mentioned above is the most mature method at present. But for large scale databases or data warehouses, it is quite difficult to store data whether in memory or hard disk. Whereas the problems exist in present FP-growth association rule mining algorithm, this paper compresses the data which provide frequent items to a FP-network, and store this FP-network by forming an association matrix. This method not only inherits the merit that FP-tree model does not produce candidates and does not visit database frequently, but also overcomes the demerit that FP-tree model is difficult to update and maintain. It is especially fit for association rule mining in large-scale databases.

# 2 Backgrounds

## 2.1 The System Construction of Data Mining

Every researcher will form his own system construction when constructing a data mining system. For example, the three logical level (namely data obtaining level, data storage level, and data mining level) defined by MSMiner system; system construction integrated by LAP and OLAM and so on.

From the respect of databases, there is a typical system construction of data mining, which is mainly combined with six parts as follows [38].

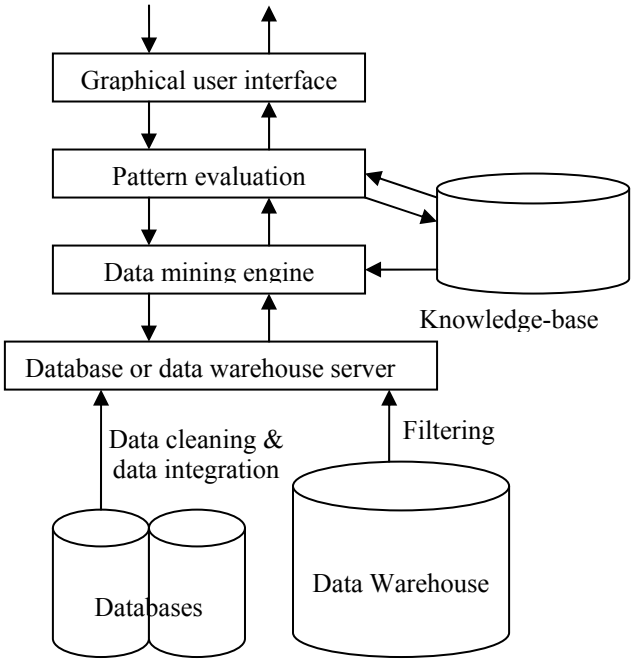


Fig. 1 Typical system structure of data mining

## 2.2 The Process of Data Mining

The process of data mining is basically similar at present algorithms. Several typical ones are listed as follows:

1) From the respect of mining environment, DMgroup [39] proposed five steps, namely determination of business object, data preparation, data mining, result analysis, and knowledge assimilation;

2) From the respect of knowledge discovery, the process of data mining is divided into three steps, namely data preparation, data mining, and result evaluation;

3) From the respect of modeling, SPSS proposed the '5A' model of data mining. The '5A' are namely Assess, Access, Analyze, Act, and Automate;

4) In other respects, SAS proposed the SEMMA model (Sample, Explore, Modify, Model, Assess), Special Interest Group on Knowledge Discovery and Date Mining proposed the CRISP-DM (Cross-Industry Standard Process for Data Mining) standard [40].

In general, according to the typical system

construction of data mining above, the process of data mining can be divided into four steps as follows:

1) Data preparation;
2) Data preprocessing;
3) Data mining;
4) Pattern evaluation and expression.

In the process above, the purpose of data preparation is to ensure the target data according to the customers' demand. Data preprocessing usually contains noise immunization, missing value estimation, repeated records elimination, and data type conversion [41].

### 2.3 The Definition of Association Analysis

Suppose $I=\{i_1 \quad i_2 \quad \cdots \quad i_m\}$ is the set of items, task concerned data D is the set of database transactions, each transaction T is also the set of items and there exists $T \subseteq I$. Suppose A is another set of items, association rule is the implication form expressed as $A \Rightarrow B$, where $A \subset I, B \subset I$ and $A \cap B = \varphi$, there exists the rule $A \Rightarrow B$ in transaction D with support value $s$, where $s$ is the percentage of $A \cup B$ within transaction D. And the confidence $c$ of rule $A \Rightarrow B$ in transaction D is defined as the percentage of transactions contain both $A$ and $B$, namely:

$$\begin{aligned} support(A \Rightarrow B) &= P(A \cup B) \\ confidence(A \Rightarrow B) &= P(B \mid A) \end{aligned} \tag{1}$$

Rules satisfy both minimum support value ( $min\_sup$ ) and minimum confidence value ( $min\_conf$ ) is defined as strong rules. The set of items is defined as itemset, itemset contains $k$ items is defined as $k-itemset$. If the itemset satisfies $min\_sup$, it is defined as frequent itemset, frequent $k-itemset$ is usually expressed as $L_k$.

### 2.4 The Property of Association Rule

In the process of knowledge discovery in data bases, association rule is the knowledge pattern which describes the probability rule of items appears at the same time.

The four basic properties of association rule are confidence, support, expected confidence and lift, which can be described as follows.

1) Confidence: the appearance probability of B in the case of A appears;
2) Support: the probability of A and B appear at the same time;
3) Expected confidence: the appearance probability of B;
4) Lift: the ratio of confidence/expected confidence.

The calculation formulas for the four properties above are listed in table 1.

Table 1 The Formulas for Four Properties

| Property | Formula |
|---|---|
| Confidence | $P(B \mid A)$ |
| Support | $P(B \cap A)$ |
| Expected confidence | $P(B)$ |
| Lift | $P(B\|A) / P(B)$ |

### 2.5 The Process of Association Rule Mining

Association rule mining is an important component of data mining. From Agrawal R. put forward association rule pattern of Boolean type in 1990s, association rule mining is now widely used in different fields.

The process of association rule mining usually contains two steps:

1) The first step is to find out all the frequent itemsets. As it is defined above, the occurrence frequency of these itemsets is no smaller than the presupposed $min\_sup$;

2) The second step is to generate strong association rules from frequent itemsets. As it is defined above, these rules must satisfy both minimum support value and minimum confidence value.

## 3 Transaction-item Association Matrix

### 3.1 Definition

Database which provides frequent itemsets is usually the association between transaction and item, there is a transaction database below, where the first arrange is transaction, TID is transaction ID, the second arrange is itemset namely which items associate with the transaction, the itemset is $[I1 \quad I2 \quad I3 \quad I4 \quad I5]$.

Table 2 Transaction Database Table

| TID | Item ID |
|---|---|
| T100 | I1,I2,I5 |
| T200 | I2,I4 |
| T300 | I2,I3 |
| T400 | I1,I2,I4 |
| T500 | I1,I3 |
| T600 | I2,I3 |
| T700 | I1,I3 |
| T800 | I1,I2,I3,I5 |

| T900 | I1,I2,I3 |
|---|---|

This transaction database can be described in the form as $D = B^{(0)} I$ :

$$\begin{bmatrix} T001 \\ T002 \\ T003 \\ T004 \\ T005 \\ T006 \\ T007 \\ T008 \\ T009 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} I1 \\ I2 \\ I3 \\ I4 \\ I5 \end{bmatrix} \quad (2)$$

Where $D$ models transaction sets; $I$ models itemset; matrix $B^{(0)}$ models the transaction-item association matrix, its element $b_{ij}$ can be defined as: to transaction $i$, if it associates with item $j$, the corresponding element will be 1; otherwise be 0.

The equation above can also be described as follows:

$$D_i = \sum_{j=1}^{M} b_{ij} I_j \quad i = 1,2,\cdots,N \quad (3)$$

Where $i = 1,2,\cdots,N$ is transaction set; $j = 1,2,\cdots,M$ is the itemset, in the example above, the corresponding values of $N$ and $M$ are $N = 9$ and $M = 6$.

### 3.2 Association Rule Mining

To the association matrix mentioned above, if the minimum support value is 2, namely $\min\_\sup = 2$; we add the arranges of the association matrix and get:

$$L_j^1 = \sum_{i=1}^{N} b_{ij}^{(0)} \quad (4)$$

Where $L_j^1$ is the frequency of the jth item appears in the whole transaction database. In the example above, $L_1^1 = 6$, $L_2^1 = 7$, $L_3^1 = 6$, $L_4^1 = 2$, $L_5^1 = 2$. And if $\min\_\sup$ is defined as the minimum support threshold value, there is:

$$L_j^1 \geq \min\_\sup \quad (5)$$

It is a frequent 1-itemset. We make elementary arrange operations to the matrix above, namely add arrange 1 to arrange 2, 3, 4…respectively; neglect arrange 1, add arrange 2 to arrange 3, 4…respectively; … And then make calculation as follows:

$$\begin{cases} 1+1=1 \\ 1+0=0 \\ 0+1=0 \\ 0+0=0 \end{cases} \quad (6)$$

It is obtained:

$$B^{(1)} I^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I1+I2 \\ I1+I3 \\ I1+I4 \\ I1+I5 \\ I2+I3 \\ I2+I4 \\ I2+I5 \\ I3+I4 \\ I3+I5 \\ I4+I5 \end{bmatrix} \quad (7)$$

Similarly, if we add every arrange of the matrix, it is obtained:

$$L_j^2 = \sum_{i=1}^{N} b_{ij}^{(1)} \quad (8)$$

Where $j = 1,2,\cdots,N^{(1)}$ is the 2-itemset. If:

$$L_j^2 \geq \min\_\sup \quad (9)$$

It should be the frequent 2-itemset. In the example above, $L_1^2 = 4$, $L_2^2 = 4$, $L_3^2 = 1$, $L_4^2 = 2$, $L_5^2 = 4$, $L_6^2 = 2$, $L_7^2 = 2$, $L_8^2 = 0$, $L_9^2 = 1$, $L_{10}^2 = 0$. The frequent 2-itemsets are $I1+I2$, $I1+I3$, $I1+I5$, $I2+I3$, $I2+I4$, $I2+I5$. Similarly, we can get frequent 3-itemsets $I1+I2+I3$, $I1+I2+I5$.

## 4 The Realization of Association Matrix Mining Algorithm

From the deducibility above, it is obvious that the association matrix mining algorithm is very intuitionistic. Compared with the Apriori algorithm, the process of repetitive database scanning to find the frequent itemset can be avoided. But if the transaction database is a large one, to put the association matrix into computer's memory entirely is also unpractical. So it is necessary to improve the efficiency of storage and operation when realizing the association matrix mining algorithm.

### 4.1 Matrix Predigestion

From the deducibility above, it is obtained that when some arrange's (itemset) count is smaller than the minimum support value, this arrange will be

added to other arranges and the result obtained is just the same. That is, the combination with other items does not gain the frequent itemsets. This rule can also be obtained from equation (4).

Therefore, after every check for frequent itemsets, infrequent itemsets will be deleted and will not be brought into the next calculation step. In the example above, association matrix $B^{(1)}$ can be transformed as follows:

$$B^{(1)}I^{(1)}=\begin{bmatrix}1&0&1&0&0&1\\0&0&0&0&1&0\\0&0&0&1&0&0\\1&0&0&0&1&0\\0&1&0&0&0&0\\0&0&0&1&0&0\\0&1&0&0&0&0\\1&1&1&1&0&1\\1&1&0&1&0&0\end{bmatrix}\begin{bmatrix}I1+I2\\I1+I3\\I1+I5\\I2+I3\\I2+I4\\I2+I5\end{bmatrix} \quad (10)$$

If we make elementary arrange operation to the association matrix above, it is obtained:

$$B^{(2)}I^{(2)}=\begin{bmatrix}0&1&0&0&1&0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\1&1&1&0&1&1&1&0&1&1&0&1&0&1&0\\1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}I1+I2+I3\\I1+I2+I5\\I1+I2+I3\\I1+I2+I4\\I1+I2+I5\\I1+I3+I5\\I1+I2+I3\\I1+I2+I3+I4\\I1+I2+I3+I5\\I1+I2+I5\\I1+I2+I3+I5\\I1+I2+I5\\I2+I3+I4\\I2+I3+I5\\I2+I4+I5\end{bmatrix} \quad (11)$$

After deleting the repetitive items and infrequent items, it is obtained:

$$B^{(2)}I^{(2)}=\begin{bmatrix}0&1\\0&0\\0&0\\0&0\\0&0\\0&0\\0&0\\1&1\\1&0\end{bmatrix}\begin{bmatrix}I1+I2+I3\\I1+I2+I5\end{bmatrix} \quad (12)$$

We make elementary transform to the association matrix, there is no frequent itemsets any more, which means the calculation ends.

## 4.2 Sparse Storage of Matrix

Deducibility above shows that the frequent 2-itemset matrix $B^{(1)}$ is larger than the frequent 1-itemset matrix $B$, and the frequent 3-itemset matrix $B^{(2)}$ is larger than the frequent 2-itemset matrix $B^{(1)}$. But we can find the rule as follows: the more combinations of items, the sparser the corresponding matrix is. There is only 16 elements valued 1 in association matrix $B^{(2)}$, account for only not more than 12% of the whole elements (9×15=135). The matrix will be sparser if the matrix is enlarged.

Therefore, as the elements of the association matrix are neither 0 nor 1, we can use sparse storage technique of matrix and it only need to store the row and arrange number of those elements valued 1. An element occupies 2 storage units, as matrix $B^{(2)}$ the storage quantity is 24% of the whole. About 76% memory can be saved, and it is evidently efficient to large-scale matrix.

## 4.3 Partitioning Calculation of Association Matrix and Data Updating

To a practical database, data accumulate ceaselessly. And due to the huge amounts storages of data, it takes much time to traverse the data base by mining association rule when new data are added, especially in large-scale data warehouses. Furthermore, it is necessary to partition the association matrix to increase the efficiency of data operations to large-scale database.

The formed association matrix $B$ can be divided into four parts as follows:

$$B=\begin{bmatrix}B_{11}&B_{12}\\B_{21}&B_{22}\end{bmatrix} \quad (13)$$

Where $B_{11}, B_{21}$ model frequent itemsets in the transaction; $B_{12}, B_{22}$ model infrequent itemsets in the transactions. In the association matrix $B^{(1)}$ above, if transactions T001~T006 is one suite, T007~T009 is another suite, the partitioning matrix can be described as follows:

$$B^{(1)}I^{(1)}=\left[\begin{array}{c|c}101001 & 0000\\ 000010 & 0000\\ 000100 & 0000\\ 100010 & 1000\\ 010000 & 0000\\ \underline{000100} & \underline{0000}\\ 010000 & 0000\\ 111101 & 0010\\ 110100 & 0000\end{array}\right]\left[\begin{array}{c}I1+I2\\ I1+I3\\ I1+I5\\ I2+I3\\ I2+I4\\ \underline{I2+I5}\\ I1+I4\\ I3+I4\\ I3+I5\\ I4+I5\end{array}\right] \quad (14)$$

This shows that the four partitioning matrix can calculate the support value respectively, the rules of calculation are as follows:

1) If the support values of every arranges of matrix $B_{11}$ is bigger than the minimum support value, the support value calculation of matrix $B_{21}$ is no more necessary.

2) If the support values of every arranges of matrix $B_{21}$ is bigger than the minimum support value, the support value calculation of matrix $B_{11}$ is no more necessary.

3) When calculating the frequent value of matrix $B_{22}$, the frequent value should be added to the frequent value of matrix $B_{21}$. Its matrix $B_{22}$ should be transferred to $B_{21}$ if the infrequent item changes into frequent item.

In practical calculation processes, it is only necessary to store the item frequent value that calculated, but not the whole matrix, thus decreasing the calculation amount obviously.

### 4.4 Compared with Apriori Algorithm

In the process of association rule mining, the forming process of association matrix $B^{(0)}\rightarrow B^{(1)}\rightarrow B^{(2)}...$ is actually same with the 'connect' step of Apriori algorithm, which is the main step to form candidate frequent itemsets. Where matrix $B^{(0)}$ is the basic transaction-item association matrix; $B^{(1)}$ is the transaction-item association matrix to form candidate frequent 1-itemset; $B^{(2)}$ is the transaction-item association matrix to form candidate frequent 2-itemset analogically. Different from Apriori algorithm, the forming of candidate itemsets is realized by elementary matrix operations, it is comparatively simple and the process is very clear.

The predigestion of transaction-item association matrix is similar with the 'cut' step of Apriori algorithm. It is to predigest the process that does not produce frequent itemsets. As the amount of frequent itemsets is less and less, the increasement of association matrix dimension $B^{(0)}\rightarrow B^{(1)}\rightarrow B^{(2)}...$ is slower and slower.

Although the process of transaction-item association matrix mining algorithm is similar with Apriori algorithm in many respects, it is obvious that the use of elementary matrix operations, sparse storage techniques and matrix partitioning techniques in the association rule mining increases the data mining efficiency obviously.

## 5 Frequent Pattern Network Algorithm

FP-tree model compresses the database which provides frequent items into a directed FP-tree, so there exist difficulties of maintaining and updating the data. To avoid such a demerit, here we propose an undirected FP-network model.

### 5.1 FP-network Model

The process of establishing the FP-network model is as follows:

1) Regard each item as a node in the network, in the example above, there are five items I1, I2, I3, I4, I5, so there are five nodes in the network;

2) Traverse the database and form the arc of the network for transaction T100, and as transaction T100 has three items, it is made up of two arcs, I1→I2, I2→I5, establish such two arcs and evaluate them as 1 respectively (the arc capability is 1); as this transaction begins at node I1, evaluate node I1 as 1 (the node capability is 1); and as it ends at node I5, evaluate node I5 as -1 (the node capability is -1, which means flow to the opposite direction);

3) Visit other transactions as the rule shown above, the FP-network established can be shown as follows.

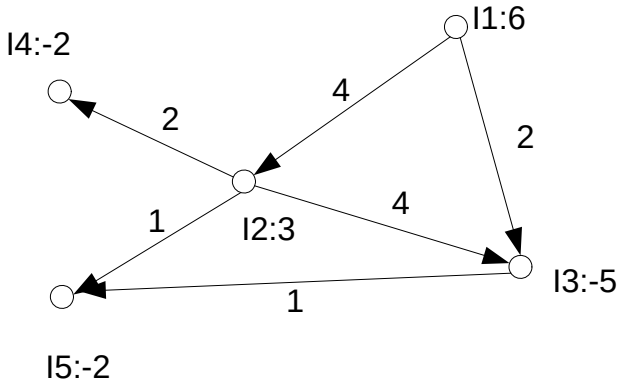


Fig. 2 FP-network Figure

The FP-network model above has such characters:

1) The sum of node capability is 0, the sum of node inject capability and node pour capability are both 9;

2) On each node, the sum of arc capability and node capability is 0. For example, to node I3, arc capability (5) adds node capability (-5) is 0;

3) The amount of node injects capability or pour capability is the frequency that the node appears in the transaction.

Compared with FP-tree model, FP-network model compresses the database which provides frequent items to a network, but this network can not express the total transaction yet. That is the network itself enlarges the amount of transaction.

Take node I5 as an example, there are two transactions associating with it, that is I1, I2, I3, I5 and I1, I2, I5, but there are four paths in the graph that associating with node I5, that is I1, I2, I3 ,I5 / I1, I2, I5 / I1, I3, I5 / I2, I5, but the latter two paths do not actually exist.

## 5.2 Association Matrix Expression of FP-network

The FP-network model above can be described as an association matrix. As the actual FP-network can not distinguish the real path, thus enlarging the transaction set. To avoid such demerits, when describing FP-network by computer, we use the expression form of transaction (path) - item (node) association matrix, which is shown as follows.

$$\begin{bmatrix} T001 \\ T002 \\ T003 \\ T004 \\ T005 \\ T006 \\ T007 \\ T008 \\ T009 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} I1: & 6 \\ I2: & 3 \\ I3: & -5 \\ I4: & -2 \\ I5: & -2 \end{bmatrix} \quad (15)$$

Where the numbers 6, 3,-5,-2,-2 correspond to items I1, I2, I3, I4, I5 present the corresponding node capability. So to store a FP-network can be conversed to store an association matrix and the capability of corresponding nodes.

## 5.3 FP-network Algorithm

It is convenient to realize association rule mining in the use of FP-network model and association matrix. The steps of association rule mining in FP-network algorithm are as follows:

1) Begin from such a node whose node capability is not positive;

2) Search for all the paths in the association matrix evaluated 1 that corresponds to this node, add the rows correspond to these paths, and evaluate the arranges 1 for those not less than min_sup; Otherwise 0;

3) Those nodes evaluated 1 in the calculation result make up the frequent itemset;

4) This process continues until all the nodes with nonpositive capability are mined.

In the example above, we mine from node I5 because its node capability is negative. The arrange corresponds to node I5 is arrange 5, and the rows whose elements are 1 are row 1 and row 8, add these two rows, we get $[2\ \ 2\ \ 1\ \ 0\ \ 2]$, as the min-support value is 2, we can transform it to $[1\ \ 1\ \ 0\ \ 0\ \ 1]$, then the corresponding node set I1, I2, I5 is a frequent itemset.

And then mine from node I4. Add row 2 and row 4 whose elements value 1 on arrange 4, we get $[1\ \ 2\ \ 0\ \ 2\ \ 0]$ and $[0\ \ 1\ \ 0\ \ 1\ \ 0]$ after transform, so their frequent itemset is I2,I4.

At last, to node I3, we add row 3,5,6,7,8,9 to get $[4\ \ 4\ \ 6\ \ 1\ \ 1]$ and $[1\ \ 1\ \ 1\ \ 0\ \ 0]$ after transform, so their frequent itemset is I1, I2, I3.

Up to now, mining ends, the combination between the frequent itemsets that mined makes all the frequent itemset.

## 5.4 Maintaining and Updating of FP-network

A main demerit of FP-tree model is the difficulties in data maintaining and updating, because when new data are added or the database is updated, the node position in FP-tree may be changed which makes the FP-tree has to be established again.

But his problem does not exist in FP-network model, because FP-network is stored in the form of association matrix and the node orders in transaction-item association matrix are discretional. For example, the transaction-item association matrix above can be changed as follows:

$$\begin{bmatrix} T001 \\ T002 \\ T003 \\ T004 \\ T005 \\ T006 \\ T007 \\ T008 \\ T009 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} I5: & 2 \\ I2: & 5 \\ I3: & 0 \\ I4: & -1 \\ I1: & -6 \end{bmatrix} \quad (16)$$

As the order of node I5 and I1 is changed; the new FP-network formed can be expressed as follows:

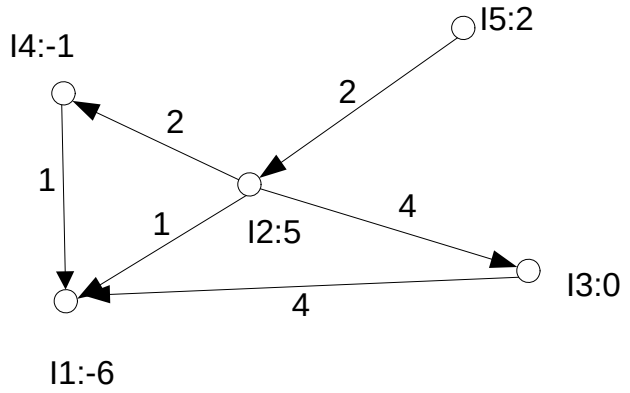


Fig. 3 FP-network Figure

The FP-network's association rule mining above begins from node I1 and gets $[2\ 4\ 4\ 1\ 6]$, $[1\ 1\ 1\ 0\ 1]$ after transform. As node I3 is a temporary node, it is neglectable. So it actually should be $[1\ 1\ 0\ 0\ 1]$, the frequent pattern is I5, I2, I1; Then mine from node I4, we get the frequent itemset I2, I4; And then mine from node I3, get the frequent item pattern I2, I3, I1. Up to now, the mining ends, we get the same result as above.

So the FP-network model expressed in the form of association matrix has no relationship with the node orders, thus getting over the difficulties of data maintaining and updating in FP-tree model.

## 6 Conclusions

This paper changes the process of association rule mining into elementary matrix operations through the establishment of transaction-item association matrix, which makes the process of association rule mining clearer and increases the efficiency of data mining.

Especially, some operation techniques of large-scale matrix such as sparse storage techniques, matrix partitioning techniques can be used into data mining, which increases the efficiency of data mining all the more.

Aiming at the demerits exist in the association rule mining algorithm at present, this paper establishes a FP-network model. FP-network model is established on the basis of the definition of node capability and arc capability. Compared with the FP-tree model, FP-network model is a connected-graph model and stored through transaction-item association matrix. We get several conclusions:

1) FP-network model compresses the data that association rule mining needed to a graph; this is similar with FP-tree model;

2) Compared with FP-tree model, FP-network model enlarges the scale of items that stored, and with the help of storing data through transaction-item association matrix, the transactions needed can be confirmed;

3) FP-network algorithm and FP-growth algorithm is similar in calculation efficiency, but FP-network algorithm is more convenient to update and maintain data that mined, thus increasing the calculation efficiency of association rule mining algorithm.

*References:*

[1] G. Gory, S. Piatetsky, *The data-mining industry coming of age*, Morgan Kaufmann Publisher, 2000.

[2] Knowledge discovery in database, *KDD in China*. Available: http://www.chinakdd.com

[3] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases, *Proc. of ACM SIGMOD International Conference on Management of Date*, Washington D C, 1993, pp. 207-216.

[4] L. Fu, Rule generation from neural networks. *IEEE Transactions on Systems*, Man, Cybernetics, Vol. 24, No. 8, 1994, pp. 1114-1124.

[5] L. Fu, A neural network model for learning domain rules based on its activation function characteristics. *IEEE Transactions on Neural Networks*, Vol. 9, No. 5, 1998, pp. 787-795.

[6] H. Tsukimoto, Extracting rules from trained neural networks. *IEEE Transactions on Neural Networks*, Vol. 11, No. 2, 2000, pp. 377-389.

[7] B. Cestnik, I. Kononenko, I. Bratko, ASSISTANT 86: a knowledge elicitation tool for sophisticated users, *Proc. of EWSL-87*, Bled, Yugoslavia, 1987, pp. 31-45.

[8] G. Pagallo, D. Haussler, Boolean feature discovery in empirical learning. *Machine Learning*, Vol. 5, 1990, pp. 71-99.

[9] C.E. Brodley, P.E. Utgoff, Multivariate decision trees. *Machine Learning*, Vol. 19, 1995, pp. 45-77.

[10] G. Deffuant, Neural units recruitment algorithm for generation of decision trees, *Proc. of International Joint Conference on Neural Networks*, San Diego, CA, 1990, 1, pp. 637-642.

[11] I.K. Sethi, Entropy nets: From decision trees to neural networks. *Proc. of the IEEE*, Vol. 78, No. 10, 1990, pp. 1605-1613.

[12] T.D. Sanger, A tree-structured adaptive network for function approximation in high-dimensional spaces. *IEEE Trans Neural Networks*, Vol. 2, No. 2, 1991, pp. 285-293.

[13] M. Kubat, Decision trees can initialize radial-basis function networks. *IEEE Trans Neural Networks*, Vol. 9, No. 5, 1998, pp. 813-821.

[14] Z. Pawlak, Rough set theory and its applications to data analysis. *Cybernetics and Systems*, Vol. 29, No. 1, 1998, pp. 661-688.

[15] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, *Slowinski Red. Intelligent Decision Support Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer Academic Publishers, 1992.

[16] M. Kryszkiewicz, Rough set approach to incomplete information systems. *Information Sciences*, Vol. 112, 1998, pp. 39-49.

[17] J.A. Michael, S. Gordon, *Data Mining Techniques*, NewYork: John Wiley&sons, 1997.

[18] D. Lewis, Training Algorithms for Linear Text Classifiers, *Proc. of 19th Annual International ACM SIGIR Conference: ACM Press*, 1996, pp. 298-306.

[19] U. Klements, Schnatting, Deep Knowledge Discovery from Natural Language Texts, *Proc. of 3rd Conference on Knowledge Discovery and Data Mining*. US: Kluwer Academic Publishers, 1997, pp. 175-178.

[20] L. Vohandu, R. Kussik, A. Torim, E. Aab, G. Lind, Some monotone systems algorithms for data mining. *WSEAS Transactions on Information Science and Applications*, Vol. 3, no. 4, 2006, pp. 802-809.

[21] K.M. Yang, E.H. Kim, S.H. Hwang, S.H. Choi, Fuzzy concept mining based on formal concept analysis. *INTERNATIONAL JOURNAL OF COMPUTERS*, Issue 3, Vol. 2, 2008, pp. 279-290.

[22] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, Fast discovery of association rules, *Fayyad U M, Piatetsky-Shapiro G, Smyth Peds. Advance in Knowledge Discovery and Data Mining*. CA: AAAIPress, 1996, pp. 307-328.

[23] R. Strikant, R. Agrawal, Mining quantitative association rules in large relational tables, *Proc. of ACM SIGMOD Conference on Management of Data (SIGMOD'96)*, Montreal, 1996, pp. 1-12

[24] R. Strikant, R. Agrawal, Mining generalized association rules, *Proc. of 21st International Conference on Very Large Data Bases*, Zurich, 1995, pp. 407-419.

[25] B. Yoon, R.C. Lacher, Extracting rules by destructive learning, *Proc. of IEEE ICNN'94*, New York: IEEE Press, 1994, pp. 1766-1771.

[26] R. Iváncsy, I. Vajk, Efficient sequential pattern mining algorithms. *WSEAS Transactions on Computers*, Vol. 4, No. 2, 2005, pp. 96-101.

[27] R. Iváncsy, I. Vajk, PD-Tree: A New Approach to Subtree Discovery, *WSEAS Transactions on Information Science and Applications*, Vol. 2, No. 11, 2005, pp. 1772-1779.

[28] R. Agrawal, R. Srikant, Fast algorithm for mining association rules, *Proc. of 20th International Conference on VLDB*, Santiago, Chile, 1994, pp. 487- 499.

[29] T. Han, M. Kamber. *Data Mining: Concepts and Techniques.* Beijing: Higher Education Press, 2001.

[30] T. Han, M. Kamber. M. Fan, X.F. Meng, *Translated Data Mining: Concepts and Techniques*. Beijing: China Machine Press, 2001(in Chinese).

[31] R. Agrawal, T.C. Shafer. *Parallel mining of association rules: Design, implementation, and experience*. IBM Research Report RJ10004, 1996.

[32] A. Savasmr, E. Omiccinski, S. Navathc, An efficient algorithm for mining association rules, *Proc. of 21th International Conference on VLDB*, Zurich, Switzerland, 1995, pp. 432- 444.

[33] M. Marian, S. Saddys, F.L. Vivian, M.J. Polo, A method for mining quantitative association rules, *Proc. of 6th WSEAS International Conference on Simulation, Modeling and Optimization*, Lisbon, Portugal, September 22-24, 2006, pp. 173-178.

[34] T.H.S. Alex, I. Maria, S. Bala, Mining infrequent and interesting rules from transaction records, *Proc. of 7th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'08)*, University of Cambridge, UK, Feb 20-22, 2008, pp. 515-520.

[35] L. Zhou, S. Yau, Efficient association rule mining among both frequent and infrequent

items. *Computer Maths Applications*, Vol.54, No.6, 2007, pp. 737-749.
[36] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, *Proc. of 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD 2000)*, New York: ACM Press, 2000, pp. 1-12.
[37] T. Han, M. Kember, *Data Mining: Concepts and Techniques*. 2nd ed. Beijing: Higher Education Press, 2001.
[38] P.R. Olivia, *Data Mining Cookbook: Modeling Data for Marketing, Risk, and Customer Relationship Management*. John Wiley & sons, Inc. 2003.
[39] Available: http://jwc.njue.edu.cn/tjx/data%20mining/webpage/DMgroup.htm
[40] Available: http://www.crisp-dm.org
[41] Q. Tong, Research on data mining in the scientific data grid, Ph.D. dissertation, Dept. Computer Science, Chinese academy of sciences, Beijing, China, 2006.