Frequent Pattern Network Mining Algorithm Based on Transaction-item Association Matrix

YI-JUN WANG¹, WEI-QING SUN², JIN-TAO SHE³, SHENG-BIAO WEI³, CHENG-MIN WANG²

¹ Department of Electrical Engineering
Northeast Dianli University
169 Changchun Road, Jilin
CHINA
wangyijun69@126.com

² Department of Electrical Engineering
Shanghai Jiaotong University
800 Dongchuan Road, Shanghai
CHINA

³ Nanping Electric Power Bureau
211 Bayi Road, Nanping, Fujian
CHINA

Abstract: - To increase the efficiency of data mining is the emphasis in this field at present. Aiming at the difficulties of data maintaining and updating in association rule mining 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, FP-network 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 letter not only inherits the merits of FP-growth algorithm, but also maintains and updates data conveniently. It improves the efficiency of association rule mining.

Key-Words: - data mining; association rule; FP-growth algorithm; FP-network algorithm

1  Introduction

Association rule is an important research subject put forward by Agrawal in reference [1]. The existing association rule mining methods are mainly Apriori algorithm and its improvements [2-6]. 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.

Therefore, J. Han put forward a method of creating frequent pattern sets through frequent pattern tree [7-8]. 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 doesn’t produce candidate generations and traverse 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 a large scale database, 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 Transaction-item Association Matrix

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 I2 I3 I4 I5].

Table 1 Transaction Database Table

<table>
<thead>
<tr>
<th>TID</th>
<th>Item ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>T100</td>
<td>I1,I2,I5</td>
</tr>
<tr>
<td>T200</td>
<td>I2,I4</td>
</tr>
<tr>
<td>T300</td>
<td>I2,I3</td>
</tr>
<tr>
<td>T400</td>
<td>I1,I2,I4</td>
</tr>
<tr>
<td>T500</td>
<td>I1,I3</td>
</tr>
<tr>
<td>T600</td>
<td>I2,I3</td>
</tr>
<tr>
<td>T700</td>
<td>I1,I3</td>
</tr>
<tr>
<td>T800</td>
<td>I1,I2,I3,I5</td>
</tr>
<tr>
<td>T900</td>
<td>I1,I2,I3</td>
</tr>
</tbody>
</table>

This transaction database can be described as
\[
D = BI
\]

Where \( D \) models transaction sets; \( I \) models itemset; matrix \( B \) is the transaction-item association matrix, the 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:
\[
D_i = \sum_{j=1}^{M} b_{ij} I_j \quad i = 1,2,\ldots,N
\]

Where \( i = 1,2,\ldots,N \) is transaction set; \( j = 1,2,\ldots,M \) is the itemset, in the example above \( N = 9 \) and \( M = 6 \).

3 Frequent Pattern Network Algorithm

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

3.1 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 above, the FP-network established can be shown as follows.

![Fig. 1 FP-network Figure](image)

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 that 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.
3.2 Association Matrix Expression of FP-network
The FP-network model above can be described as association matrix. As the actual FP-network cannot distinguish the real path, thus enlarging the transaction set. To avoid such demerits, when describe FP-network by computer, we use transaction (path)-item (node) association matrix shown as follows:

\[
\begin{bmatrix}
T001 & 1 & 1 & 0 & 0 & 1 \\
T002 & 0 & 1 & 0 & 1 & 0 \\
T003 & 0 & 1 & 1 & 0 & 0 \\
T004 & 1 & 1 & 0 & 1 & 0 \\
T005 & 1 & 0 & 1 & 0 & 0 \\
T006 & 0 & 1 & 1 & 0 & 0 \\
T007 & 1 & 0 & 1 & 0 & 0 \\
T008 & 1 & 1 & 1 & 0 & 1 \\
T009 & 1 & 1 & 1 & 0 & 0 \\
\end{bmatrix}
\]

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.

3.3 FP-network Algorithm
It’s convenient to realize association rule mining in the use of FP-network model and association matrix. The steps of association rule mining are as follows:
1) Begin from the 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.

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. This 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:

\[
\begin{bmatrix}
T001 & 1 & 1 & 0 & 0 & 1 \\
T002 & 0 & 1 & 0 & 1 & 0 \\
T003 & 0 & 1 & 1 & 0 & 0 \\
T004 & 1 & 1 & 0 & 1 & 0 \\
T005 & 1 & 0 & 1 & 0 & 0 \\
T006 & 1 & 1 & 0 & 0 & 0 \\
T007 & 0 & 1 & 0 & 1 & 0 \\
T008 & 1 & 1 & 0 & 1 & 0 \\
T009 & 1 & 1 & 0 & 0 & 0 \\
\end{bmatrix}
\]

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

![Fig. 2 FP-network Figure](image-url)

The FP-network’s association rule mining above begins from node I1 and get \([2 \ 4 \ 4 \ 1 \ 6]\) and \([1 \ 1 \ 1 \ 0 \ 1]\) 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.

5 Conclusion
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 is more convenient to update and maintain data that mined, thus increasing the calculation efficiency of association rule mining algorithm.

6 Acknowledgments
The authors thank for the supports from HI-TECH RESEARCH AND DEVELOPMENT PROGRAM OF CHINA (No. 2006AA05Z214).

References: