ASSOCIATION RULE MINING BASED ON TRADE LIST

Ms. Sanober Shaikh¹ Ms. Madhuri Rao²

¹Department of Information Technology, TSEC, Bandra (w), Mumbai
ºDepartment of Information Technology, TSEC, Bandra (w), Mumbai

ABSTRACT

In this paper a new mining algorithm is defined based on frequent item set. Apriori Algorithm scans the database every time when it finds the frequent item set so it is very time consuming and at each step it generates candidate item set. So for large databases it takes lots of space to store candidate item set. In undirected item set graph, it is improvement on apriori but it takes time and space for tree generation. The defined algorithm scans the database at the start only once and then from that scanned data base it generates the Trade List. It contains the information of whole database. By considering minimum support it finds the frequent item set and by considering the minimum confidence it generates the association rule. If database and minimum support is changed, the new algorithm finds the new frequent items by scanning Trade List. That is why it’s executing efficiency is improved distinctly compared to traditional algorithm.

KEYWORDS

Undirected Item set Graph, Trade List

1. INTRODUCTION

Mining Association rule is very important field of research in data mining. The problem of mining Association rule is put forward by R.S Agarwal first in 1993. Now the Association rules are widely applied in E-commerce, bank credit, shopping cart analysis, market analysis, fraud detection, and customer retention, to production control and science exploration. etc. [1]

Now a days we will find many mining methods for finding the frequent item set such as Apriori algorithm, Frequent Pattern-Tree algorithm etc. Apriori algorithm’s disadvantage is it generates lot of candidate itemsets and scans database every time. If database contains huge number of transactions then scanning the database for finding the frequent itemset will be too costly and it generates a lot of candidates. Next FP-Tree algorithm’s advantage is it does not produce any candidate items but it scans database two times in the memory allowed. But when the memory does not meet the need, this algorithm becomes more complex. It scans the database more than two times and the I/O expenses will increase [2]. That is why there is need to design an efficient algorithm which updates, protects and manages the association rule in large transactional database. So far many researchers made analysis and research for how to efficiently update the association rules and put forward corresponding algorithm. There are two instances in the problem of Association Rule update. The first instance is when the database is changed then how to find frequent item sets. FUFIA Algorithm is the representational updating method for this problem. The second instance is when the minimum support is changed then how to find frequent items sets. IUA algorithm is the representational updating method for this problem. These updating algorithms have both advantages and disadvantages. This paper proposes a dynamic algorithm of frequent mining based on undirected item set graph which scans the database only once and then saves the information of original database in undirected item set
graph and finds the frequent item sets directly from the graph. It does not generate any candidate items. When database and minimum support is changed, the algorithm rescans the undirected item set graph to get the new frequent item sets.[3]

2. B.A.SIC C.ONCEPT O.F A.SSOCIATION RULE

Association rule finds interesting associations and/or correlation relationships among large set of data items. Association rule shows attribute value conditions that occur frequently together in a given dataset. A typical and widely-used example of association rule mining is Market Basket Analysis.

For example, data are collected using bar-code scanners in supermarket. Such `market basket’ databases consist of a large number of transaction records. Each record lists all items bought by a customer on a single purchase transaction. Managers would be interested to know if certain groups of items are consistently purchased together. They could use this data for adjusting store layouts (placing items optimally with respect to each other), for cross-selling, for promotions, for catalog design and to identify customer segments based on buying patterns.

Association rules do not represent any sort of causality or correlation between the two item sets. The problem of mining association rules can be described as below: if \( I = \{I_1, I_2, \ldots, I_n\} \) is the set of items. Suppose \( D \) is database transaction set and each transaction \( T \) contains set of items, such that \( T \subseteq I \). Each transaction has identifier called as TID i.e. transaction id. Suppose \( A \) is a set of items and transaction \( T \) is said to contain \( A \) only if \( A \subseteq T \).

Association rule is an implication like as \( A \implies B \) in which \( A, B \subseteq I \) and \( A \cap B = \emptyset \). [6]

Definition of support: The support is the percentage of transactions that demonstrate the rule. An item set is called frequent if its support is equal or greater than an agreed upon minimal value – the support threshold. [8]

Definition of Confidence: Every association rule has a support and a confidence.

An association rule is of the form: \( X \Rightarrow Y \).

\( X \Rightarrow Y \): if someone buys \( X \), he also buys \( Y \).

The confidence is the conditional probability that, given \( X \) present in a transition, \( Y \) will also be present. Confidence measure, by definition:

\[
\text{Confidence}(X \Rightarrow Y) = \frac{\text{support}(X, Y)}{\text{support}(X)}
\]

The aim of association rule is to find all association problems having support and confidence not less than given threshold value. For the given support i.e. minsupp, if the item set of \( D \)’s support is not less than minsupp, then it can say that \( D \) is the frequent item set.

3. F.I.NDING F.REQUENT I.TEM S.ETS

First step is to scan the database. It makes each item as a node and at the same time it makes the supporting trade list for each node. Supporting trade list is a binary group \( T=\{\text{Tid, Itemset}\} \) (where Tid is transaction id and Itemset is trade item set). Given database that includes five items and nine transactions (shown in table one). Suppose that minimum support minsupp is two. Table two contains the information of support trade list of table one.

With this Trade List directly we will get information of which items are appearing in which transactions. So here number of transactions related to that item will decide count of that item. So we have count of \( I_1 \) as 6 as shown in Table 2. Similarly we will get the count of all the items in the database. Now after considering the minimum support from user we will compare that minimum support with the count. If it is greater those will be considered as frequent-1 item set.
Table 1: A Store Business Data

| TID    | The List Of 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          |

Table 2: Trade List of Commodity Item

| Commodity Item | Support Trade List                      |
|----------------|----------------------------------------|
| I1             | T100, T400, T500, T700, T800, T900     |
| I2             | T100, T200, T300, T400, T600, T800, T900 |
| I3             | T300, T500, T600, T700, T800, T900     |
| I4             | T200, T400                           |
| I5             | T100, T800                           |

In next step for finding frequent itemset do intersection of I1 and I2. In result if we will get some transactions we will get common then it means that the item is related to other transaction also. Count the numbers of those common transactions that will give the count of those two items that are bought together that many numbers of times. Example I1 ∩ I2 will get the count as 4 that means I1 and I2 are together 4 number of times in the database. Compare this with minimum support. Then we will get frequent-2 itemset. Similarly the procedure is iteratively applied.

3.1.3. Optimize Mining Algorithm Based On Undirected Item Sets Graph

The algorithm in this paper uses the search strategy of Depth first- Search to set universal undirected item graph. The specific steps are shown as follows:

Select a node \( V_i \) from node set \( V \). If the number of times \( V_i \) appears in the database is not less than the minimum support mnsupp, then \( \{V_i\} \) will belong to the item in frequent 1-item set. If count of node \( V_i \) adjacent to node \( V_j \)'s side is not less than support \( S \), then \( \{V_i, V_j\} \) will belong to the item in frequent 2-item set. When there are three nodes in undirected item set graph and count of each side of the node is not less than minimum support mnsupp, these three nodes \( \langle V_k, V_m, V_n \rangle \) will belong to frequent 3-item set. When there more than three nodes in undirected item sets graph then count of each side of the node should not be less than minimum support mnsupp and all the subset of these \( n \) nodes should be frequent.
3.2. Updating Trade List

When database and minimum support i.e. minsupp is changed the Trade List should be changed accordingly. If we want to add some new items to the database, then Trade List is updated accordingly.

3.2.1. Database Affair Changed

For example, when a new item T910 is added to table one; the result is as shown as in table three.

| TID  | The list of items |
|------|------------------|
| 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       |
| T910 | I1, I4           |

A new item T910 have added at this time. So the arisen number of side <I1, I4> is two. As shown in fig.1, frequent 1-item set is L1= {I1, I2, I3, I4, I5}; frequent 2-item set is L2= { {I1, I2}, {I1, I3}, {I2, I3}, {I2, I4}, {I2, I5}, {I1, I4} }; frequent 3-item set is L3= { {I1, I2, I3}, {I1, I2, I5}, {I1, I2, I4} }.

3.2.2 Minimum support changed

For example, when the minimum support minsupp is three, frequent 1-item set= {I1, I2, I3}; frequent 2- item is L2= { {I1, I2}, {I1, I3}, {I2, I3} }.
4. RESULTS

4.1 Results of Apriori Algorithm

Fig 1: Frequent Item Set with Apriori Algorithm with database shown in Table 1

| 1-I1 | 2-I2 | 3-I3 | 4-I4 | 5-I5 | 6-I1,I2 | 7-I1,I3 | 8-I1,I4 | 9-I1,I5 | 10-I2,I5 |
|-----|-----|-----|-----|-----|--------|--------|--------|--------|---------|
|     |     |     |     |     |        |        |        |        |         |

4.2 Results of Trade

Fig 2: Main Form

In Fig 2 form the first i.e. Item Set File asks for the database from which you want to retrieve the frequent items. Here for input of Item set file one .isf file is made as shown in Fig 3. In that file the code for connectivity with database is made. Through the code the database is converted to a text file. In the first line write name of .isf file that will be converted to a format which the code will accept.

Fig 3: Item Set File
When we will click on Generate button in Fig 2, Trade list is made from which we can come to know how many number of items are present in input database as shown in Fig 4.
Then with the help of this Trade list we will get frequent items easily.
Here minimum support is 3. Now the count of each item is compared with minimum support. If count is greater than minimum support those items will be frequent item sets as shown in fig 5.
| File | Edit | Format | View | Help |
|------|------|--------|------|------|

Fig 5: Frequent Items

1-11
2-14
3-15
4-16
5-17
6-111
7-110
8-125
9-12
10-13
11-18
12-110
13-113
14-114
15-120
16-19
17-115
18-19
19-122
20-112
21-118
22-124
23-130
24-117
25-10
26-127
27-121
28-11,14
29-11,15
30-11,17
31-11,11
32-11,10
33-11,12
34-11,13
35-14,15
36-14,16
37-14,12
38-14,18
39-14,114
40-14,112
41-15,17
42-15,111
43-15,125
44-15,12
45-16,12
46-16,13
47-16,18
48-16,110
49-16,114
50-16,122
51-16,112
52-15,118
53-15,124
54-17,111
55-17,110
| Frequent Items |
|----------------|
| 56-17, i9     |
| 37-11, i19    |
| 58-11, t25    |
| 59-11, i2     |
| 60-11, i3     |
| 61-11, i8     |
| 62-11, i13    |
| 63-11, i9     |
| 64-11, i9     |
| 65-12, i3     |
| 66-12, i8     |
| 67-12, i10    |
| 68-12, i13    |
| 69-12, i14    |
| 70-12, i9     |
| 71-12, i15    |
| 72-12, i22    |
| 73-12, i12    |
| 74-12, i18    |
| 75-12, i24    |
| 76-13, i8     |
| 77-13, i10    |
| 78-13, i13    |
| 79-13, i10    |
| 80-18, i14    |
| 81-18, i12    |
| 82-110, i14   |
| 83-110, i20   |
| 84-110, i15   |
| 85-110, i22   |
| 86-110, i12   |
| 87-110, i16   |
| 88-114, i22   |
| 89-114, i12   |
| 90-114, i13   |
| 91-114, i24   |
| 92-19, i27    |
| 93-19, i21    |
| 94-115, i22   |
| 95-115, i12   |
| 96-115, i18   |
| 97-122, i12   |
| 98-122, i18   |
| 99-122, i24   |
| 100-112, i18  |
| 101-112, i24  |
| 102-112, i30  |
| 103-118, i24  |
| 104-118, i30  |
| 105-124, i30  |
| 106-14, i5, i11 |
| 107-16, i2, i14 |
| 108-12, i8, i14 |
| 109-12, i14, i4 |
| 110-12, i12, i4 |
### Frequent Items

| Fig 5 (cont): Frequent Items |
|-------------------------------|
| File | Edit | Format | View | Help |
|------|------|--------|------|------|
| 111  | i18  | i12    | i14  |      |
| 112  | i17  | i11    | i15  |      |
| 113  | i11  | i12    | i15  |      |
| 114  | i12  | i13    | i15  |      |
| 115  | i12  | i18    | i16  |      |
| 116  | i12  | i10    | i16  |      |
| 117  | i12  | i14    | i16  |      |
| 118  | i12  | i22    | i16  |      |
| 119  | i12  | i12    | i16  |      |
| 120  | i12  | i18    | i16  |      |
| 121  | i18  | i14    | i16  |      |
| 122  | i14  | i22    | i16  |      |
| 123  | i14  | i12    | i16  |      |
| 124  | i14  | i18    | i16  |      |
| 125  | i14  | i24    | i16  |      |
| 126  | i22  | i12    | i16  |      |
| 127  | i22  | i18    | i16  |      |
| 128  | i22  | i24    | i16  |      |
| 129  | i12  | i18    | i16  |      |
| 130  | i12  | i24    | i16  |      |
| 131  | i18  | i24    | i16  |      |
| 132  | i1  | i11    | i17  |      |
| 133  | i1  | i19    | i17  |      |
| 134  | i11  | i19    | i17  |      |
| 135  | i11  | i19    | i17  |      |
| 136  | i11  | i19    | i11  |      |
| 137  | i13  | i18    | i11  |      |
| 138  | i13  | i10    | i12  |      |
| 139  | i18  | i14    | i12  |      |
| 140  | i18  | i12    | i12  |      |
| 141  | i10  | i14    | i12  |      |
| 142  | i10  | i12    | i12  |      |
| 143  | i10  | i18    | i12  |      |
| 144  | i14  | i12    | i12  |      |
| 145  | i14  | i18    | i12  |      |
| 146  | i12  | i18    | i12  |      |
| 147  | i12  | i22    | i10  |      |
| 148  | i22  | i12    | i10  |      |
| 149  | i22  | i18    | i10  |      |
| 150  | i12  | i18    | i10  |      |
| 151  | i12  | i22    | i14  |      |
| 152  | i22  | i24    | i14  |      |
| 153  | i22  | i12    | i14  |      |
| 154  | i22  | i18    | i14  |      |
| 155  | i22  | i24    | i14  |      |
| 156  | i12  | i18    | i14  |      |
| 157  | i12  | i24    | i14  |      |
| 158  | i18  | i24    | i14  |      |
| 159  | i12  | i18    | i15  |      |
| 160  | i12  | i12    | i12  |      |
| 161  | i12  | i18    | i12  |      |
| 162  | i12  | i24    | i12  |      |
| 163  | i12  | i18    | i12  |      |
| 164  | i12  | i24    | i12  |      |
| 165  | i18  | i24    | i12  |      |
Confidence of each item is compared with minimum confidence given by user and strong association rule is formed. The items having confidence greater than or equal to minimum confidence, are stored in file shown in Fig 6.
Fig 6: Association Rule

| Rule          | Confidence |
|---------------|------------|
| I7 → I1       | 80%        |
| I9→I1         | 75%        |
| I4→I1         | 100%       |
| I4→I2         | 77.78%     |
| I6→I2         | 60%        |
| I2→I4         | 60%        |
| I2→I14        | 60%        |
| I8→I12→I4     | 100%       |
| I8→I12→I4     | 100%       |
| I7→I15        | 60%        |
| I11→I5        | 62.5%      |
| I35→I5        | 100%       |
| I7→I11→I5     | 60%        |
| I11→I25→I5    | 100%       |
| I6→I2         | 62.5%      |
| I22→I16       | 66.67%     |
| I6→I12        | 62.5%      |
| I6→I18        | 62.5%      |
| I2→I3         | 71.43%     |
| I2→I14        | 60%        |
| I2→I16        | 60%        |
| I2→I18        | 60%        |
| I8→I14→I6     | 100%       |
| I4→I22→I6     | 75%        |
| I4→I18→I6     | 75%        |
| I4→I14→I6     | 75%        |
| I22→I16       | 80%        |
| I22→I18       | 80%        |
| I22→I14→I6    | 75%        |
| I16→I12, I18  | 62.5%      |
| I12→I18→I16   | 71.43%     |
| I12→I14→I6    | 75%        |
| I21→I22→I6    | 75%        |
| I12→I18→I6    | 75%        |
| I12→I24→I6    | 75%        |
| I12→I14→I6    | 75%        |
| I18→I12→I18   | 75%        |
| I12→I22→I18   | 75%        |
| I12→I24→I18   | 75%        |
| I12→I14→I18   | 75%        |
| I18→I12→I14→I6| 75%        |
| I18→I12→I14→I6| 75%        |
| I18→I12→I14→I6| 75%        |
**Fig 7(cont): Association Rule**

| Rule            | Confidence |
|-----------------|------------|
| $I_7 \rightarrow I_{11}$ | 100%       |
| $I_{11} \rightarrow I_7$   | 62.5%      |
| $I_7 \rightarrow I_{19}$   | 60%        |
| $I_{19} \rightarrow I_7$   | 75%        |
| $I_7 \rightarrow I_9$      | 60%        |
| $I_7 \rightarrow I_{11}, I_{11}$ | 80%    |
| $I_1, I_{11} \rightarrow I_7$ | 100%   |
| $I_7 \rightarrow I_{11}, I_{19}$ | 60%   |
| $I_{11}, I_{19} \rightarrow I_7$ | 100%   |
| $I_7 \rightarrow I_{11}, I_9$ | 60%        |
| $I_7 \rightarrow I_{11}, I_9$ | 100%       |
| $I_7 \rightarrow I_{11}, I_{11}, I_{11}$ | 60%   |
| $I_1, I_{19}, I_{11} \rightarrow I_7$ | 100%   |
| $I_{19} \rightarrow I_{11}$ | 75%        |
| $I_{23} \rightarrow I_{11}$ | 100%       |
| $I_3 \rightarrow I_{11}$   | 80%        |
| $I_1, I_{19} \rightarrow I_{11}$ | 100%   |
| $I_3, I_{18} \rightarrow I_{11}$ | 100%   |
| $I_{19} \rightarrow I_9$   | 75%        |
| $I_8 \rightarrow I_2$      | 62.5%      |
| $I_{13} \rightarrow I_2$   | 60%        |
| $I_{14} \rightarrow I_2$   | 62.5%      |
| $I_{22} \rightarrow I_2$   | 66.67%     |
| $I_{13} \rightarrow I_{10}$ | 75%       |
| $I_{18} \rightarrow I_{12}$ | 100%      |
| $I_{12} \rightarrow I_2$   | 75%        |
| $I_{10}, I_{14} \rightarrow I_2$ | 100%   |
| $I_{10}, I_{12} \rightarrow I_2$ | 100%   |
| $I_{10}, I_{18} \rightarrow I_2$ | 100%   |
| $I_{14}, I_{18} \rightarrow I_2$ | 75%    |
| $I_{12}, I_{18}, I_{10} \rightarrow I_2$ | 100%   |
| $I_{12}, I_{18}, I_{14} \rightarrow I_2$ | 75%    |
| $I_{13} \rightarrow I_3$   | 60%        |
| $I_{20} \rightarrow I_{10}$ | 66.67%    |
| $I_{22} \rightarrow I_{10}$ | 75%       |
| $I_{22}, I_{12} \rightarrow I_{10}$ | 60%   |
| $I_{22}, I_{18} \rightarrow I_{10}$ | 60%   |
| $I_{12}, I_{22} \rightarrow I_{10}$ | 75%    |
| $I_{18}, I_{22} \rightarrow I_{10}$ | 75%    |
| $I_{12}, I_{18}, I_{22} \rightarrow I_{10}$ | 60%   |
| $I_{12}, I_{18}, I_{12} \rightarrow I_{10}$ | 75%    |
| $I_{12} \rightarrow I_{14}$ | 66.67%     |
| $I_{14} \rightarrow I_2$   | 75%        |
| $I_{14} \rightarrow I_{24}$ | 62.5%      |
| $I_{22} \rightarrow I_{14}$ | 75%        |
| $I_{12} \rightarrow I_{14}$ | 100%       |
| $I_{22}, I_{12} \rightarrow I_{14}$ | 80%   |
| $I_{22}, I_{18} \rightarrow I_{14}$ | 80%   |
| $I_{12}, I_{24} \rightarrow I_{14}$ | 100%  |
| $I_{14} \rightarrow I_{12}, I_{24}$ | 62.5%   |
| $I_{12}, I_{24} \rightarrow I_{14}$ | 71.43%   |
| $I_{12}, I_{12} \rightarrow I_{14}$ | 75%    |
| $I_{12}, I_{18}, I_{22} \rightarrow I_{14}$ | 75%   |
Fig 7(cont): Association Rule

| Rule | Confidence |
|------|------------|
| I2, I24, I22->I14 | 100% |
| I12, I18, I22->I14 | 80% |
| I12, I24, I22->I14 | 100% |
| I18, I24, I22->I14 | 100% |
| I2, I24, I12->I14 | 100% |
| I18, I24, I12->I14 | 66.67% |
| I2, I18, I2, I22->I14 | 100% |
| I12, I24, I12, I22->I14 | 100% |
| I18, I24, I12, I22->I14 | 75% |
| I12, I24, I12, I22->I14 | 100% |
| I18, I24, I12, I22->I14 | 100% |
| I12, I24, I18, I22->I14 | 100% |
| I2, I24, I18, I12->I14 | 100% |
| I2, I24, I18, I12, I22->I14 | 100% |
| 127->I9 | 75% |
| I21->I9 | 100% |
| I22->I12 | 83.33% |
| I22->I18 | 83.33% |
| I22->I124 | 66.67% |
| I12->I12 | 66.67% |
| I12->I22 | 80% |
| I22->I18 | 66.67% |
| I12->I22 | 100% |
| I12->I122 | 100% |
| I12->I12 | 83.33% |
| I12->I12 | 71.43% |
| I12->I124 | 66.67% |
| I12->I12 | 66.67% |
| I12->I12, I12, I12 | 66.67% |
| I12, I18, I2->I12 | 100% |
| I12, I24, I12->I12 | 100% |
| I12->I18, I24, I12 | 66.67% |
| I18, I24, I12->I12 | 66.67% |
| I12, I24, I18->I12 | 100% |
| I12, I24, I18, I12->I12 | 100% |
| I18->I12 | 77.78% |
| I24->I12 | 77.78% |
| I2, I24->I12 | 100% |
| I18, I24->I12 | 85.71% |
| I18, I30->I12 | 75% |
| I24, I30->I12 | 80% |
| I12, I24, I18->I12 | 100% |
| I24, I30, I18->I12 | 75% |
| I18->I24 | 77.78% |
| I24->I18 | 77.78% |
| I2, I24->I18 | 100% |
| I2, I24->I18 | 80% |
| I30->I24 | 71.43% |
| I2, I22->I12, I18 | 100% |
| I2, I22, I6->I12, I18 | 100% |
| I2, I22, I10->I12, I18 | 100% |
| I2, I22, I14->I12, I18 | 100% |
5. CONCLUSION
In this project candidate items are not generated. The information of items of original database is saved in undirected item set graph. Then the information of frequent item set is found by searching trade list. The apriori scans the database too many times and generating candidates in each step. If we have huge amount of data then scanning such data and storage of huge amount of candidates is very difficult. Algorithm based on “A new association rule mining based on undirected itemset graph” having the disadvantage of tree generation. It takes time for generation tree. Now Trade list technique as compare to apriori and undirected itemset graph takes less amount of time and give the proper results.

6. ACKNOWLEDGEMENT
1. Ms Madhuri Rao (Guide)
2. Mr. Naushad Shaikh

7. REFERENCES
[1] S. Chai, J. Yang, Y. Cheng, “The Research of Improved Apriori Algorithm for Mining Association Rules”, 2007 IEEE.
[2] S. Chai, H. Wang, J. Qiu, “DFR: A New Improved Algorithm for Mining Frequent Item sets”, Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)
[3] R. Agrawal, T. Imielinski, A. Swami, “Mining Association Rules between Sets of Items in Very Large Databases [C]”, Proceedings of the ACM SIGMOD Conference on Management of Data, Washington, USA, 1993-05: 207-216
[4] R. Agrawal, T. Srikant, “Fast Algorithms for Mining Association Rules in Large Database [C]”, Proceedings of 20th VLDB Conference, Santiago, Chile, 1994: 487-499
[5] L Guan, S Cheng, and R Zhou, “Mining Frequent Patterns without Candidate Generation [C]”, Proceedings of SIGMOD’00, Dallas, 2000:1-12.
[6] Dongme Sun, Shaohua Teng, Wei Zhang, “An algorithm to improve the effectiveness of Apriori”, Proceedings of 6th IEEE International Conference on Cognitive Informatics (ICCI07), IEEE2007.
[7] http://en.wikipedia.org/wiki/Apriori_algorithm.