ASSOCIATION RULE MINING BASED ON TRADE LIST

Ms. Sanober Shaikh[1] Ms. Madhuri Rao[2]

[1]Department of Information Technology, TSEC, Bandra (w), Mumbai
s.sanober1@gmail.com
[2]Department of Information Technology, TSEC, Bandra (w), Mumbai
my_rao@yahoo.com

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

DOI : 10.5121/ijdkp.2011.1603

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. Basic Concept Of Association 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 $A \Rightarrow B$ in which $A, B \subset I$ 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 => Y.

X => 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:

Confidence(X=>Y) = support(X, Y)/ 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. Finding Frequent Item Sets

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= {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 I1 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. 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.

Table 3: The New Data in a Store

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 <*I*1, *I*4> 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 },{ I1 , I5},{ I2 , I3 },{ I2 , I4 },
{I2, I5}, {I1, I4 }}; frequent 3-item set is L3={{ I1, I2, I3 },{ I1, I2, I5},{ I1, I2, I4}}.

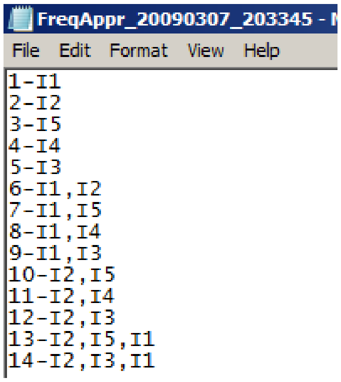
3.2.2 Minimum support changed

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

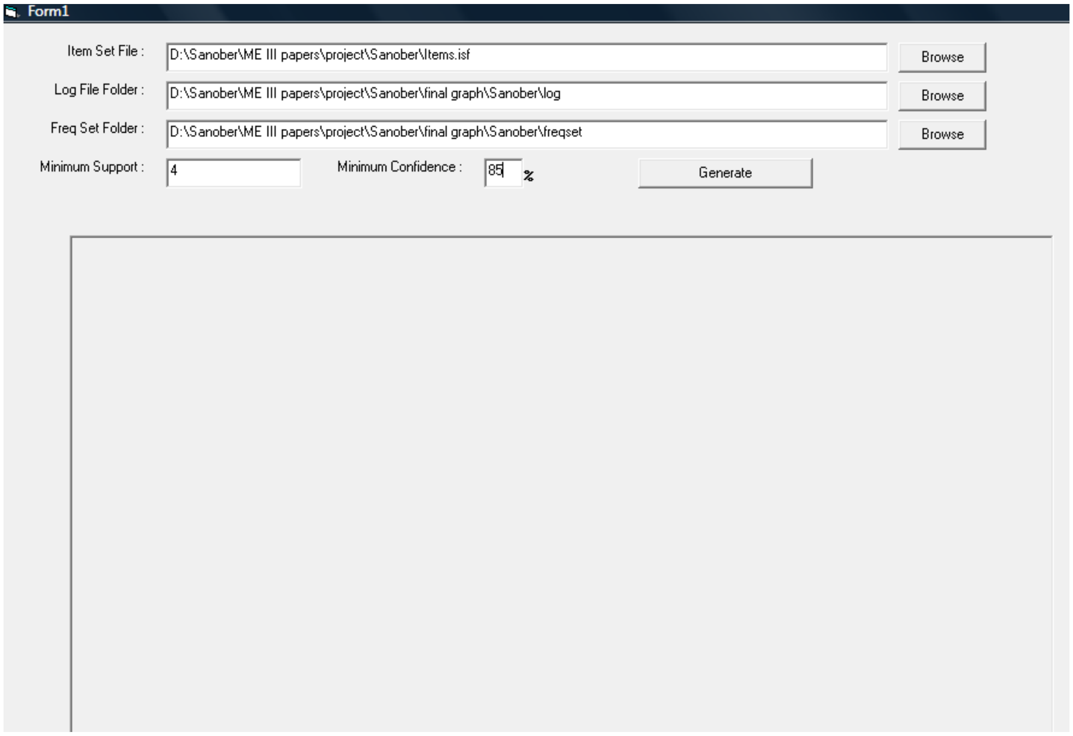
Fig1: Frequent Item Set with Apriori Algorithm with database shown in Table 1



```
1-I1
2-I2
3-I5
4-I4
5-I3
6-I1,I2
7-I1,I5
8-I1,I4
9-I1,I3
10-I2,I5
11-I2,I4
12-I2,I3
13-I2,I5,I1
14-I2,I3,I1
```

4.2 Results of Trade List

Fig 2: Main Form



Item Set File : D:\Sanober\ME III papers\project\Sanober\Items.isf

Log File Folder : D:\Sanober\ME III papers\project\Sanober\final graph\Sanober\log

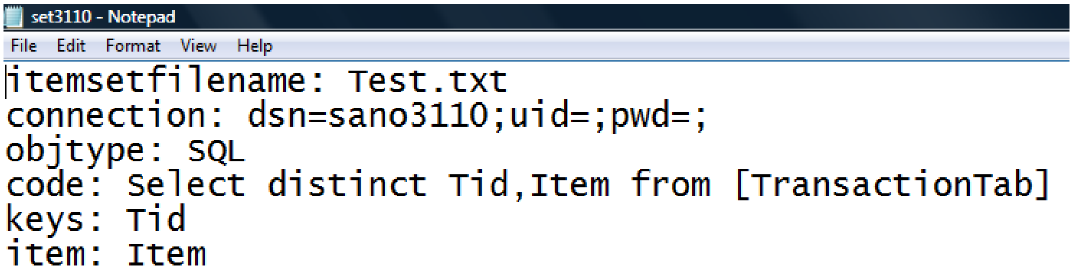
Freq Set Folder : D:\Sanober\ME III papers\project\Sanober\final graph\Sanober\freqset

Minimum Support : 4

Minimum Confidence : 85 %

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.

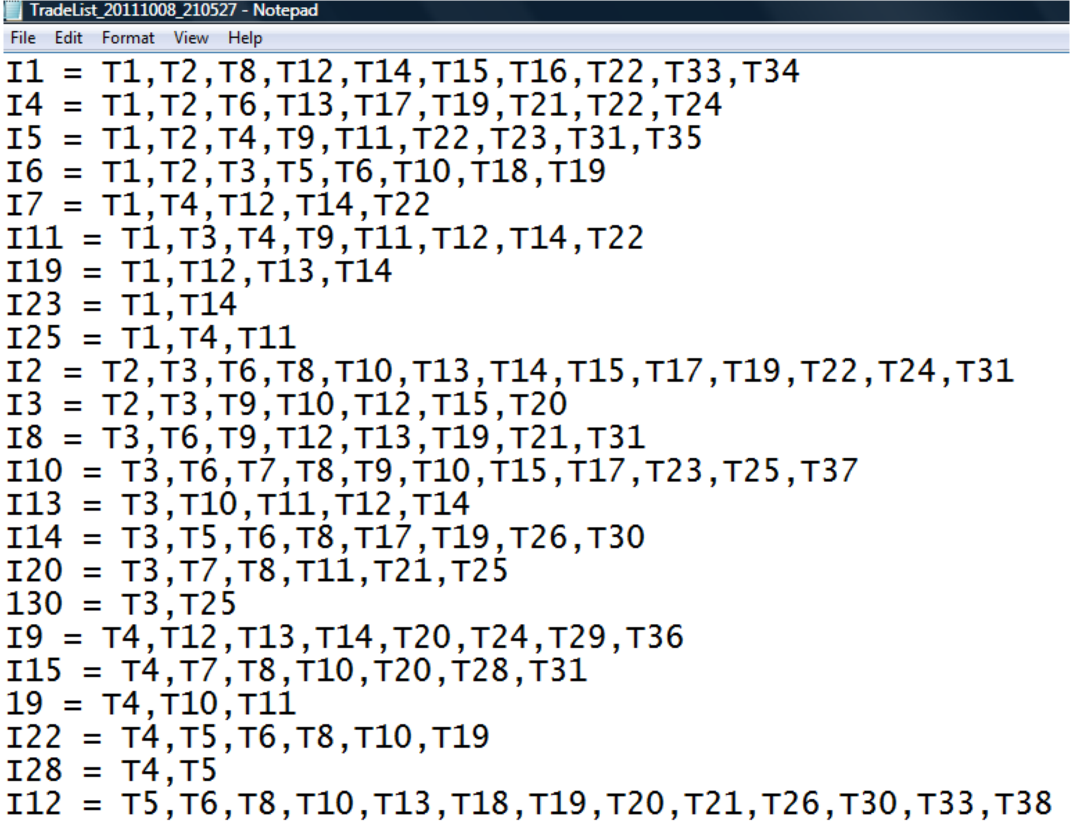
Fig 3: Item Set File



```
itemsetfilename: Test.txt
connection: dsn=sano3110;uid=;pwd=;
objtype: SQL
code: Select distinct Tid,Item from [TransactionTab]
keys: Tid
item: Item
```

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.

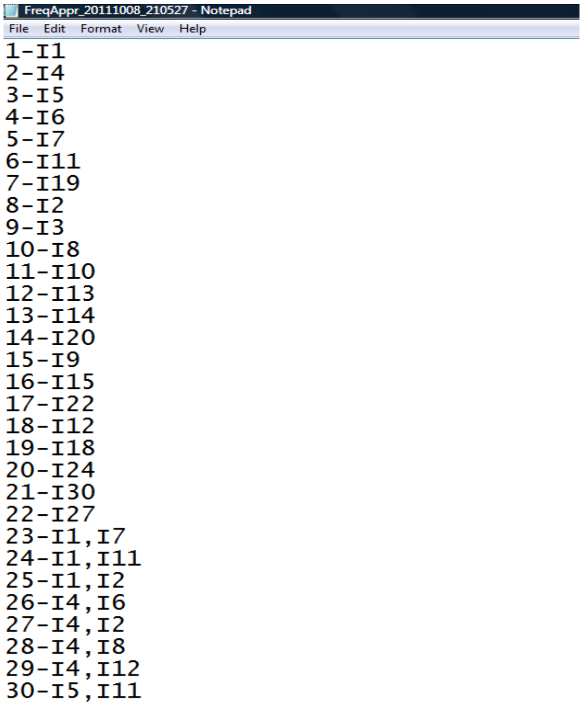
Fig 4: Trade List


TradeList_20111008_210527 - Notepad
File Edit Format View Help

```
I1 = T1,T2,T8,T12,T14,T15,T16,T22,T33,T34
I4 = T1,T2,T6,T13,T17,T19,T21,T22,T24
I5 = T1,T2,T4,T9,T11,T22,T23,T31,T35
I6 = T1,T2,T3,T5,T6,T10,T18,T19
I7 = T1,T4,T12,T14,T22
I11 = T1,T3,T4,T9,T11,T12,T14,T22
I19 = T1,T12,T13,T14
I23 = T1,T14
I25 = T1,T4,T11
I2 = T2,T3,T6,T8,T10,T13,T14,T15,T17,T19,T22,T24,T31
I3 = T2,T3,T9,T10,T12,T15,T20
I8 = T3,T6,T9,T12,T13,T19,T21,T31
I10 = T3,T6,T7,T8,T9,T10,T15,T17,T23,T25,T37
I13 = T3,T10,T11,T12,T14
I14 = T3,T5,T6,T8,T17,T19,T26,T30
I20 = T3,T7,T8,T11,T21,T25
130 = T3,T25
I9 = T4,T12,T13,T14,T20,T24,T29,T36
I15 = T4,T7,T8,T10,T20,T28,T31
19 = T4,T10,T11
I22 = T4,T5,T6,T8,T10,T19
I28 = T4,T5
I12 = T5,T6,T8,T10,T13,T18,T19,T20,T21,T26,T30,T33,T38
```

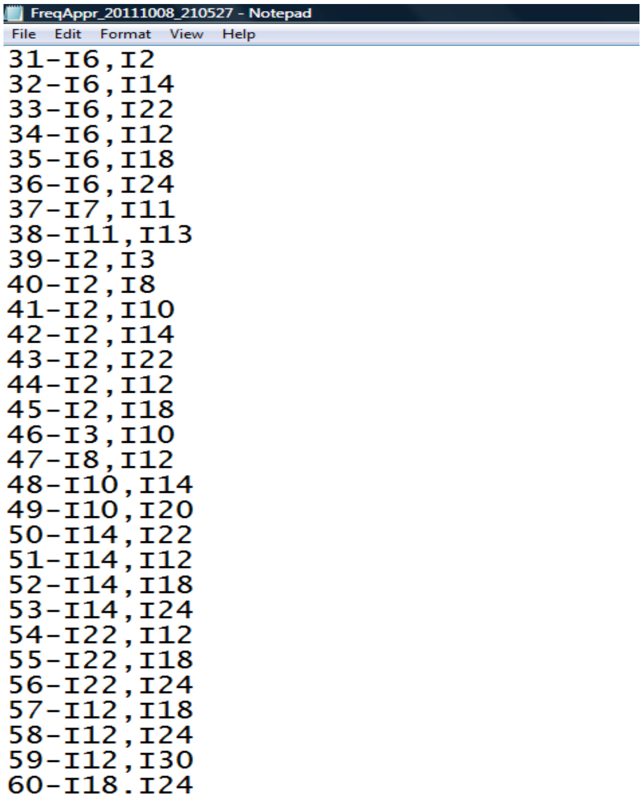
Then with the help of this Trade list we will get frequent items easily.Here minimum support is 3. Now the count of each item is compared with minimum support. If count is greater than minimum support those items will be frequent item sets as shown in fig 5.

Fig 5: Frequent Items



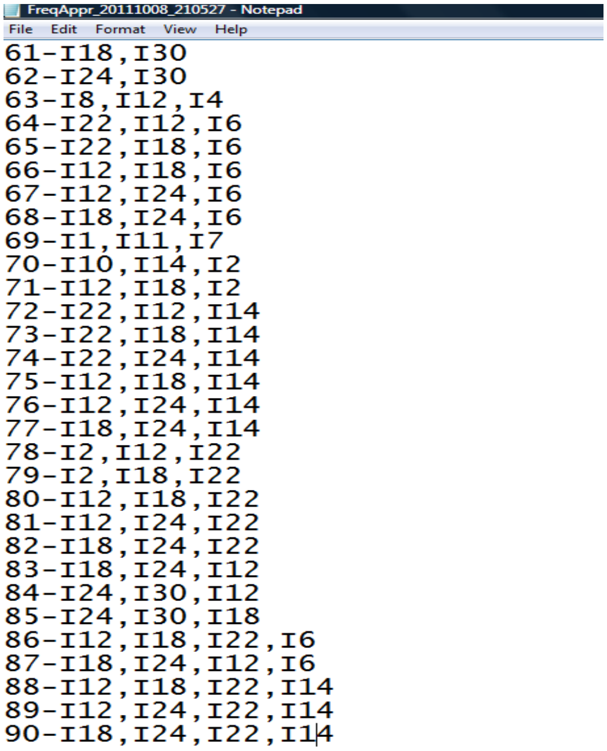
```
1-I1
2-I4
3-I5
4-I6
5-I7
6-I11
7-I19
8-I2
9-I3
10-I8
11-I10
12-I13
13-I14
14-I20
15-I9
16-I15
17-I22
18-I12
19-I18
20-I24
21-I30
22-I27
23-I1,I7
24-I1,I11
25-I1,I2
26-I4,I6
27-I4,I2
28-I4,I8
29-I4,I12
30-I5,I11
```

Fig 5 (cont): Frequent Items



```
31-I6,I2
32-I6,I14
33-I6,I22
34-I6,I12
35-I6,I18
36-I6,I24
37-I7,I11
38-I11,I13
39-I2,I3
40-I2,I8
41-I2,I10
42-I2,I14
43-I2,I22
44-I2,I12
45-I2,I18
46-I3,I10
47-I8,I12
48-I10,I14
49-I10,I20
50-I14,I22
51-I14,I12
52-I14,I18
53-I14,I24
54-I22,I12
55-I22,I18
56-I22,I24
57-I12,I18
58-I12,I24
59-I12,I30
60-I18.I24
```

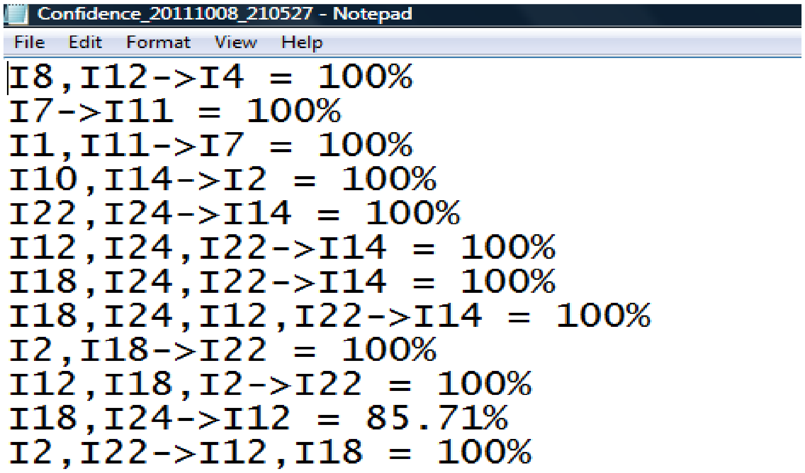
Fig 5 (cont): Frequent Items


FreqAppr_20111008_210527 - Notepad
File Edit Format View Help

```
61-I18,I30
62-I24,I30
63-I8,I12,I4
64-I22,I12,I6
65-I22,I18,I6
66-I12,I18,I6
67-I12,I24,I6
68-I18,I24,I6
69-I1,I11,I7
70-I10,I14,I2
71-I12,I18,I2
72-I22,I12,I14
73-I22,I18,I14
74-I22,I24,I14
75-I12,I18,I14
76-I12,I24,I14
77-I18,I24,I14
78-I2,I12,I22
79-I2,I18,I22
80-I12,I18,I22
81-I12,I24,I22
82-I18,I24,I22
83-I18,I24,I12
84-I24,I30,I12
85-I24,I30,I18
86-I12,I18,I22,I6
87-I18,I24,I12,I6
88-I12,I18,I22,I14
89-I12,I24,I22,I14
90-I18,I24,I22,I14
```

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


Confidence_20111008_210527 - Notepad
File Edit Format View Help

```
I8,I12->I4 = 100%
I7->I11 = 100%
I1,I11->I7 = 100%
I10,I14->I2 = 100%
I22,I24->I14 = 100%
I12,I24,I22->I14 = 100%
I18,I24,I22->I14 = 100%
I18,I24,I12,I22->I14 = 100%
I2,I18->I22 = 100%
I12,I18,I2->I22 = 100%
I18,I24->I12 = 85.71%
I2,I22->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 generating 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 (ICCI'07), IEEE2007.

[7] *http://en.wikipedia.org/wiki/Apriori_algorithm*.