An Effective Method for Association Rule Mining based on Transactional Matrix

Harpreet Singh
Department of Computer Science and Engineering
National Institute of Technology
Jalandhar, India

Renu Dhir
Department of Computer Science and Engineering
National Institute of Technology
Jalandhar, India

ABSTRACT
Focus of this paper is to present a new method based on transactional matrix for finding association rules more efficiently. Apriori algorithm is one of the classical algorithms for finding association rules, but it has limitations of number of times database scanned is too large and number of candidate itemsets generated is large.To reduce these two limitations a new method tries to find the association rules directly from a matrix which is generated from the transactional database .

Keywords— Apriori algorithm, Association rule, Frequent itemsets, Transactional matrix

1. INTRODUCTION
Association rule mining [1] finds the frequent patterns, associations, correlations or casual structures among the sets of items of transactional database, relational database, or other information repositories. Association rule [1] specifies the interesting relationship between different data elements of the database or datasets. An association rule is of the form:

$X \rightarrow Y$ [Support= s%, Confidence=c%] where

1. Support s, is the probability that rule contains {X, Y}

Support ($X \rightarrow Y$) = P (XUY),

2. Confidence c, is the conditional probability that specify the c% of the transactions of database considered must specify $X \rightarrow Y$

Confidence($X \rightarrow Y$) =P(Y/X)=P(XUY)/support_count(X)

Minimum Support and Minimum Confidence are needed to eliminate the unimportant association rules. The association rule holds iff it has the support and confidence value greater than or equal to minimum support and minimum confidence threshold value.

APRIORI algorithm [2] proposed by R. Agrawal and R. Srikant is one of the classical algorithms for finding frequent itemsets and then generating association rules from these itemsets. However, Apriori algorithm has the limitation of producing a large number of candidate itemsets and scanning the database too many times. Many researchers have given different approaches for improving the performance of Apriori algorithm. Changsheng Zhang and Jing Ruan [3] had worked on the improvement of Apriori algorithm by applying dataset reduction method and by reducing the I/O spending. Changsheng and Jing Ruan applied the modified algorithm for instituting cross selling strategies of the retail industry and to improve the sales performance. Wanjun Yu, Xiaochun Wang and et.al [4] proposed a novel algorithm called as Reduced Apriori Algorithm with Tag (RAAT), which improves the performance of Apriori algorithm by reducing the number of frequent itemset generated in pruning operation, by applying transaction tag method. Dongme Sun, Sheohue Teng and et.al [5] presented a new technique based on forward and reverse scan of database. It produces the frequent itemsets more efficiently if applied with certain satisfying conditions. Sixue Bai, Xinxi Dai [6] presented a method called P-matrix algorithm to generate the frequent itemsets. It is found that the P-Matrix algorithm is more efficient and fast algorithm than Apriori algorithm to generate frequent itemsets.

In this paper, a new method based on transactional matrix is presented to find the association rules from the large transactional database. In this approach a transactional matrix is generated directly from the database and then frequent itemsets and support of each frequent itemset is generated directly from the transactional matrix . It is found that the new proposed approach finds the frequent itemsets more efficiently. The performance of new method is compared with that of Apriori algorithm with the help of an example.

2. DESCRIPTION OF THE CLASSICAL APRIORI ALGORITHM
Apriori algorithm employs an iterative approach known as level-wise search, where k-itemsets are used to explore (k+1)-itemsets. First, the set of frequent 1-itemsets L1 is found. Next, L1 is used find the set of frequent 2-itemsets L2. Then L2 is used to find the set of frequent 3-itemsets L3.The method iterates like this till no more frequent k-itemsets are found.

Apriori algorithm finds the frequent itemsets from candidate itemsets. It is executed in two steps: first, it retrieves all the frequent itemsets from the database by considering those itemsets whose support is not smaller than the minimum support (min_sup). Secondly, it generates the association rules satisfying the minimum confidence (min_conf) from the frequent itemsets generated in first step. The first step consists of join and pruning action. While joining, the candidate set Ck is produced by joining Lk-1 with itself and pruning the candidate sets by applying the Apriori property i.e. All the non-empty subsets of a frequent itemset must also be frequent.

The pseudo code for generation of frequent itemsets is given below.

```
Ck: Candidate itemset of size k
Lk: Frequent itemset of size k
{
    L1= frequent 1-itemset
    For (k=1; k! =NULL; k++)
     {
       Ck+1=Join Lk with Lk to generate Ck+1;
        Lk+1= Candidate in Ck+1 with support greater than
                or equal to min support;
     }
    End;
```

Return Lk;
}

An example of All Electronics transactional database D [1] is presented below in TABLE I to specify the process of Apriori Algorithm. Let minimum support value be 2 (min_sup=2) and minimum confidence (min_conf) be 50% (min_conf=50%).

The process of generating frequent itemsets by Apriori algorithm is shown below in Figure 1.

TABLE I. ALL ELECTRONICS TRANSACTIONAL DATABASE

TID	Itemsets
T001	I1, I2, I5
T002	I2, I4
T003	I2, I3
T004	I1, I2, I4
T005	I1, I3
T006	I2, I3
T007	I1, I3
T008	I1, I2, I3, I5
T009	I1, I2, I3

C1		L1	
Itemsets	Support	Itemsets	Support
I1	6	I1	6
I2	7	I2	7
I3	6	I3	6
I4	2	I4	2
I5	2	I5	2
C2		**L2**	
Itemsets	Support	Itemsets	Support
I1, I2	4	I1, I2	4
I1, I3	4	I1, I3	4
I1, I4	1	I1, I5	2
I1, I5	2	I2, I3	4
I2, I3	4	I2, I4	2
I2, I4	2	I2, I5	2
I2, I5	2		
I3, I4	0		
I3, I5	1		
I4, I5	0		
C3		**L3**	
Itemsets	Support	Itemsets	Support
I1, I2, I3	2	I1, I2, I3	2
I1, I2, I5	2	I1, I2, I5	2

Figure 1. Generation of frequent itemsets by applying Apriori Algorithm

In Apriori algorithm it is found that in the join step, Lk+1 is produced from Ck+1, which is produced by joining of Lk with itself. So it may produce large number of candidate itemsets. For example if there are 10^4 frequent 1-itemsets Apriori algorithm may produce 10^7 candidate2-itemsets while performing join operation [1]. So with large database the number of candidate itemsets generated by Apriori algorithm is too large and it scans the database too many times.

3. PROPOSED ALGORITHM

In this method the frequent itemsets are generated directly from the matrix which is generated from the transactional database. The matrix formed in this method is called as Transactional Matrix T shown below in Figure 2, where T1, T2, T3, Ti, Tn represents the various transactions; I1, I2, I3, Ik, Im represents the various items occurring in the transactional database. Rep represents the number of times the transaction is repeated in the transactional database. The entries y_{11}, $y_{12, y13}$ …… and so on represent either 1 or 0. If a transaction contains the item mentioned in the column then the value of that item in the corresponding row is written as 1 otherwise 0.

The proposed algorithm uses two properties:
1. All the non empty subsets of a frequent itemset must also be frequent.
So, there is no need to consider those frequent itemsets which are having non frequent subsets.
2. Number of times transaction repeated in the database is represented by the count in the repetition column of the Transactional matrix.
So, there is no need to add the repeated transaction again into the transactional matrix.

	I1	I2	I3	Ik….	Im	rep
T1	y_{11}	y_{12}	y_{13}	y_{1k}	y_{1m}	r_1
T2	y_{21}	y_{22}	y_{23}	y_{2k}	y_{2m}	r_2
…	….	…..	…..	….	…..	…
Ti	y_{i1}	y_{i2}	y_{i3}	y_{ik}	y_{im}	r_i
…						
Tn	y_{n1}	y_{n2}	y_{n3}	y_{nk}	y_{nm}	r_n

Figure 2. Generation of Transactional matrix T

The steps of proposed algorithm are as follows:
1.First scan the database to find the different items occurring in the database and then make the transactional matrix by writing all the transactions along the row side and all the items occurring in the database along the column side. Don't repeat the transaction in the transactional matrix. If a transaction is occurring more than once in the transactional database then update the rep column of the transactional matrix accordingly with respect to the particular transaction. If the transaction doesn't repeat then rep i.e. repetition value for the transaction is set to 0.

2. Now complete the Transactional matrix, if the transaction contains the item mentioned in the column then write 1 otherwise 0 in the row corresponding to that transaction.

3. The candidate set C1 is generated directly from the transactional matrix and the support count value is counted by counting the occurrence of particular item in the transactional matrix. If a transaction has rep value 1 or some value other than zero i.e. transaction is repeated more than once in the database then support count must be incremented by rep value for that particular row item.

4. For generation of L1, use C1. Move all those transactions from C1 to L1 whose support count value is not less than min_support.

5. Now for the generation of C2, consider the transactional matrix again. Scan each row of the transactional matrix in such a way so as to generate 2-itemsets by considering the combinations of two items out of those items of the row which have value of 1. Write all those 2-itemsets in the candidate itemset table C2. Then find the support count of each 2-itemset generated by using transactional matrix. Then move only those itemsets from C2 to L2 whose support count value is not less than minimum support.

6. Similarly generate L3, L4….and so on.
NOTE: While generating itemsets, it is also considered not to count the support of those itemsets which are not frequent and exclude them from candidate set straight way.

4. COMPARISON OF THE PROPOSED ALGORITHM

Comparison between the Apriori algorithm and new proposed method is presented with the help of an example assumed above. The formation of the transactional matrix T and generation of frequent itemsets by the proposed method is shown below in Figure 3 and Figure 4.

	I1	I2	I3	I4	I5	rep
I1 I2 I5	1	1	0	0	1	0
I2 I4	0	1	0	1	0	0
I1 I2 I4	1	1	0	1	0	0
I1 I3	1	0	1	0	0	1
I2 I3	0	1	1	0	0	1
I1 I2 I3 I5	1	1	1	0	1	0
I1 I2 I3	1	1	1	0	0	0

Figure 3 Generation of Transactional matrix T

Items combination	**C1**		**L1**	
Itemsets	Itemsets	Support	Itemsets	Support
I1	I1	6	I1	6
I2	I2	7	I2	7
I3	I3	6	I3	6
I4	I4	2	I4	2
I5	I5	2	I5	2
Items combination	**C2**		**L2**	
Itemsets	Itemsets	Support	Itemsets	Support
I1, I2	I1, I2	4	I1, I2	4
I1, I5	I1, I5	2	I1, I5	2
I2, I5	I2, I5	2	I2, I5	2
I2, I4	I2, I4	2	I2, I4	2
I1, I4	I1, I4	1	I1, I3	4
I1, I3	I1, I3	4	I2, I3	4
I2, I3	I2, I3	4		
I3, I5	I3, I5	1		
Items combination	**C3**		**L3**	
Itemsets	Itemsets	Support	Itemsets	Support
I1, I2, I5	I1, I2, I3	2	I1, I2, I5	2
I1, I2, I4	I1, I2, I5	2	I1, I2, I3	2
I1, I2, I3				
I2, I3, I5				
I1, I3, I5				
Items combination	**C4**		**L4**	
Itemsets	Itemsets	Support	Itemsets	Support
I1, I2, I3, I5				

Figure 4 Generation of Frequent itemsets by applying New Method

In this method first transactional matrix T is generated as shown above in Figure 3. It is found that transactional database shown in TABLE I have same set of items in {T003, T006} and {T005, T007} therefore they are considered only once in the transactional matrix T and their respective repetition value is set to 1 as they have one repetition in there occurrence . Now when ever any itemset of these transactions is considered their support count value is incremented by the value of repetition counter. The generation of frequent itemsets is shown in Figure 4. For C1 the items combinations {I1}; {I2}; {I3}; {I4}; and {I5} are considered and then their respective support count is counted by using transactional matrix T. It is found that all the items in C1 have the support count more than min_sup. Therefore, Move all the items of C1 to L1. For generation of L2 ,scan every row of the transactional matrix and consider all the 2-itemsets combinations of the elements which have value 1 in the rows. Then count the support for each itemset and move only those itemsets from C2 to L2 whose support is not less than min_sup. Therefore, itemsets {I1, I4}; {I3, I5} are not moved to L2. Next, Consider all the 3-itemsets combinations of the items having value 1 in the rows.The various combinations possible are {I1, I2, I5}; {I1, I2, I4}; {I1, I2, I3}; {I2, I3, I5}; {I1, I3, I5}. Now, by using first property i.e. all the non empty subsets of a frequent itemset must also be frequent,it is found that itemsets {I1, I2, I4};{I2, I3, I5};{I1, I3, I5} contain subsets which are not frequent. Therefore these itemsets are not included in C3. C3 will contain {I1, I2, I3} and {I1, I2, I5}. Then count the support of these itemsets by using transactional matrix T.Similarly, 4-itemsets combinations possible is considered i.e. {I1, I2, I3, I5}. This itemset doesn't satisfy the first property of this method. Therefore C4=NULL and L4= NULL. Hence all the frequent itemsets are generated.

5. CONCLUSION

The association rule mining is the process of finding out relationship between data from a large existing database. Apriori algorithm suffers from two limitations of large number of candidate itemsets generation and database is scanned too many times. The proposed new method based on transactional matrix provided in this paper solves both these problems of Apriori algorithm. It helps in mining association rules efficiently. In this new method once the transactional matrix is generated it is easy to generate the frequent itemsets directly from the transactional matrix.

6. REFERENCES

[1] Jiawei Han and Micheline Kamber, (2001), "Data mining Concepts and Techniques", Morgan kaufman academic press

[2] R.Agrawal, "Mining association rules between sets of items in large databases", Proceeding of the 1993 ACM SIGMOD conference, Washington, pp.207-216

[3] Changsheng Zhang and Jing Raun, (2009), "A Modified Apriori Algorithm with its application in Instituting Cross-Selling strategies of the Retail Industry", pp.515-518

[4] Wanjun Yu, Xiachun Wang and et.al, (2008), "The Research of Improved Apriori Algorithm for Mining Association Rules", pp. 513-516

[5] Dongme Sun and et.al, (2007), "An algorithm to improve the effectiveness of Apriori Algorithm", In Proc. 6th ICE Int. Conf. on Cognitive Informatics", pp.385-390

[6] Sixue Bai, Xinxi Dai, (2007), "An efficiency Apriori algorithm:P_matrix algorithm", First International Symposium on Data, Privacy and E-Commerce, pp.101-103