A Frequent Data Mining Technique for Transactional Data

P Geetha
Assistant Professor,
Department of Computer Science
Dr.Umayal Ramanathan College for Women,
Karaikudi-3.

***Abstract*--In Data Mining research extracting frequent itemset has been considered as an important task. Apriori is a classical algorithm for mining frequent itemset. But it is not efficient in number of datasets scanning. Based on this algorithm, this paper proposed a modified algorithm of Apriori that improves the efficiency by reducing the wasted time in a number of database scanning. The paper explained this concept with an example.**

I. INTRODUCTION

Associative rules are one of the main techniques of Data mining. In Day to day activities the volume of data increased dramatically for many technologies to help in business such as cross marketing, inventory control, finding faults in telecom network, Basket data analysis promotion assortment. The aim of data mining process is to extract information from a dataset and transform it into an understandable structure.

Associative rule is mainly used to discover frequent itemsets. The data have traditionally focused on identifying the relationship between item telling some aspect of human behavior, usually buying behavior for determining items that customer buys together.

Apriori algorithm represents the candidate generative approach. It generates candidate (k+1) itemsets based on frequent k-itemsets.

II. MODIFIED ALGORITHM OF APRIORI

In Apriori algorithm we are getting a number of iterations. But using this algorithm the number of iterations are reduced and extract the frequent itemset from the largest database. Based on this, it is possible to reduce the time consumed in transaction scanning for candidate itemset. When the k-itemset increases, the transaction between modified Apriori and original Apriori increases from the view of time consuming.The following steps are needed to extract the frequent itemsets in given time.

Step 1: Scan all the transactions
Step 2: Generate a table, L1
Step 3: calculate size of the transaction (SOT) for each transaction, count the support for each item and keep the transaction ID.
Step 4: Construct candidate item set of self going (C)
Step 5: Get the desired item set based on SOT.
Step 6: Use L to identify the target transactions for C
Step 7: Scan the target transactions to generate CK.

A. An example of modified Algorithm

Suppose we have id transaction and the minimum support the transaction set is shown in the following table1.
Table:1

Tid	items
T1	I1,I2,I5
T2	I2,I4
T3	I2
T4	I2,I4
T5	I3
T6	I1,I2,I4
T7	I1,I3
T8	I2,I3
T9	I2
T10	I1,I2,I3,I5
T11	I1,I2,I3
T12	I1,I3

Initially then calculate SOT for each transaction that is as follows.
Table:2

Tid	I1	I2	I3	I4	I5	SOT
T1	1	1	0	0	1	3
T2	0	1	0	1	0	2
T3	0	1	0	0	0	1
T4	0	1	0	1	0	2
T5	0	0	1	0	0	1
T6	1	1	0	1	0	3
T7	1	0	1	0	0	2
T8	0	1	1	0	0	2
T9	0	1	0	0	0	1
T10	1	1	1	0	1	4
T11	1	1	1	0	0	3
T12	1	0	1	0	0	2

Then, scan all transactions to get frequent 1–itemset which contains items support count and the transaction id (transit) and eliminate the in frequent itemset (support less than minimum support)

Table:3

Items	Support	T IDS
I1	6	T1,T6,T7,T10, T11,T12
I2	9	T1,T2,T3,T4,T6,T8,T9,T10,T11
I3	6	T5,T7,T8,T10.T11,T12
I4	3	T2,T4,T6
I5	2	T1,T10(Deleted)

The next step is to generate 2-itemset candidate set from L, Select the transaction based on SOT. Transaction size which is greater than or equal to 2 are considered in this time.

Tid	I1	I2	I3	I4	I5	SOT
T1	1	1	0	0	1	3
T2	0	1	0	1	0	2
T4	0	1	0	1	0	2
T6	1	1	0	1	0	3
T7	1	0	1	0	0	2
T8	0	1	1	0	0	2
T10	1	1	1	0	1	4
T11	1	1	1	0	0	3
T12	1	0	1	0	0	2

So if SOT <2 are eliminated from the list now totally 9 transactions are available the list reduces the no% of scans (transaction t3,t5 and t9 are removed)

Table:4

Items	Support	Min	T id
I1,I2	4	I1	T1,T6,T7,T10, T11,T12
I1,I3	4	I1	T1,T6,T7,T10, T11,T12
I1,I4	1	I4	T2,T4,T6 (Deleted)
I2,I3	3	I3	T5,T7,T8,T10,T11,T12
I2,I4	2	I4	T2,T4,T6 (Deleted)
I3,I4	-	I5	T1,T10 (Deleted)

Then generate the 2- item set based on the following table the removed the transaction id is available in ID‘s list there is no need to check that transaction it will reduce the number of scanning.

The same thing to generate 3- item set depending on the modified SOT table here it considered the transaction which is greater than or equal to 3(SOT>=3)

Table:5

TID	I1	I2	I3	I4	I5	SOT
TI	1	1	0	0	1	3
T2	0	1	0	1	0	2 (deleted)
deT3	0	1	0	0	0	1 (deleted)
T4	0	1	0	1	0	2 (deleted)
T5	0	0	1	0	0	1 (deleted)
T6	1	1	0	1	0	3
T7	1	0	1	0	0	2 (deleted)
T8	0	1	1	0	0	2 (deleted)
T9	0	1	0	0	0	1(deleted)
T10	1	1	1	0	1	4
T11	1	1	1	0	0	3
T12	1	0	1	0	0	2 (deleted)

Then generate 3-itemset bend on the transaction ID’s

Item	Support	Min	T_ID’s
I1,I2,I3	2	I1	T1,T6,T7,T10,T11,T12
I1,I2,I4	1	I4	T2,T4,T6
I1,I3,I4	-	I4	T2,T4,T6
I2,I3,I4	-	I4	T2,T4,T6

Here it removes all the transactions because support is less than equal to 2.

III. CONCLUSION

The typical Apriori algorithm has some bottleneck in performance for reduced the no of transaction to be scanned so it needs to optimize the algorithm. This paper proposed a new modified algorithm for overcoming this problem.

The performance of modified algorithm is optimized and can be extracted the knowledge from large database faster. This paper proposed an idea for reducing the number of scanning. In future, it will be implemented and check the performance with typical algorithm.

REFERENCES

[1] Rakesh Agarwal, Ramakrishnan srikant, Fast algorithms for mining Association Rules, proceeding of the 20th VLDB conference Santiago, chile, 1994.

[2] Rakesh Agarwal, Tomas Zlmielinski, Arun Swami , Mining Association Rules between sets of items in large Databases, proceedings of the 1993 ACM SIGMOD conference Washington DC, USA, May1993.

[3] Anurag Chouboy, Ravindra patel, J.L. Rana, “A Survey of efficient algorithms and New approach for fast discovery of frequent item set for Association Rule Mining”.

[4] Dr.E.Ramaraj, K.Kavitha,“Transaction Reduction in Actionable Pattern Mining for High voluminous datasets based on Bitmap and Class Labels” – International Journal on Computer Science and Engineering.

[5] Bayardo, R.J., ‘Efficiently mining long patterns fromdatabases’. Proceedings of the1998 ACM SIGMODInternationalConferenceonManagement of Data, June 1-4, 1998. New York, USA., pp: 85-93.

[6] Bucila, C., J. Gehrke, D. Kifer and W.White, ‘DualMiner: A dual pruningalgorithmforitemsets with constraints’.Data Mining andKnowledgeDiscovery, 72003. 241-272.

[7] Calders, T. and B. Goethals, ‘Depth-firstnon-derivable itemset mining’. Proc. SIAMInt. Conf. Data Min., 119 2005. pp: 250-261.

[8] Cheung, D.W., J. Han, V.T. Ng and C.Y.Wong, ‘Maintenance of discoveredassociation rules in large databases: Anincremental updating technique’.Proceedings of International Conference on Data Engineering, Feb. 26-Mar. 1, 1996.New Orleans, Louisiana, pp: 106-114.

[9] Park, J.S., Chen M.S.and Yu, P.S. ‘Efficientparallel mining for association rules’.Proceedings of the 4th International Conference on Information and KnowledgeManagement, Nov. 29-Dec. 2, 1995.Baltimore, MD., pp: 31-36.