Matrix-Over-Apriori: An Improvement Over Apriori Using Matrix

Vartika Mohan*, Dharmveer Singh Rajpoot

Department Of Computer Science Engineering
Jaypee Institute Of Information Technology, Noida, India
vartikamohan2@gmail.com
drdharmveer16382@gmail.com

Abstract

Despite of being one of the oldest and most popularly used algorithms for Association Rule Mining Apriori Algorithm contains many loopholes such as frequently scanning of database, generating large number of candidate- key, in addition to all these Apriori Algorithm also consumes very large amount of storage space for its processing. The major cause of its lagging behind is that it is not that much efficient in its working when dealing with large datasets. Thus to solve limitations of Apriori Algorithm, we are presenting an algorithm i.e. Matrix-Over-Apriori. This algorithm is an improvement over Apriori Algorithm which is supported by matrix and its performing various procedures over that. Also later on it is observed that our proposed algorithm is efficient in both ways i.e. space and time. Also it decreases the amount of candidate keys that is produced during entire processing. In addition to all this in our paper we have also compared Matrix-Over-Apriori with all other existing techniques for Association Rule Mining.

Keywords- AND Operation, Matrix, Apriori Algorithm, Candidate Keys.

I. INTRODUCTION

Mining association rule technique is predominantly used for market basket analysis, its objective is to find regular pattern present in shopping by customers in markets, on-line shops and mail-order companies etc. With the application of Apriori algorithm one's aim is to deduce a way by which the shop-keeper become able by simply looking in shopping cart he/she can check availability of certain products with high probability. This type of data deduced from association rules, is used to enhance the amount of sold items, by accordingly keeping the products in the shelves of a supermarket. The common application areas of association rule learning are as Web usage mining, Intrusion detection, Continuous production, Bio-informatics, Web based education System and Detecting frequent purchasing pattern by various people etc. the well known existing methods are as Apriori Algorithm, FP Growth Algorithm, AprioriTid Algorithm, Apriori hybrid Algorithm, Tertius Algorithm and AprioriHC Algorithm. Apriori, algorithm in spite of being very efficient, suffers from a number of inefficiencies or demerits which have given rise to many other algorithms. Assumes transaction database is residing in memory only. Dealing with large number of candidates is very expensive. Mining large dataset is a very complex procedure as it it requires large number of scans and that too again and again.

There are several mining algorithms of association rules. One of the most popular algorithms is Apriori whose aim is to search for frequent item sets from huge database and getting the association rule for discovering the knowledge. The limitation of the Apriori algorithm waste time for scanning the whole database searching on the frequent item sets. Our research objective is to improve Apriori by reducing that wasted time depending on scanning only some transactions. Machine learning approach works with all the possible combination of attribute values into a separate class, deduce learn rules from the left-over attributes as input and then evaluating them for support and confidence. Problem: Computation of Apriori is very interactive with the presence of large no. of classes that results in increased no. of rules).Association rule uses lower number of attributes and the algorithm i.e. Apriori is more efficient. Implementation of Apriori Algorithm is very easy on large datasets.

II. LITERATURE SURVEY

Jha et.al (2013)[1], In this paper, various different approaches for frequent data mining has been compared using association rules. They have emphasized to further extend their work of including redundant set theory with apriori algorithm to improved frequent pattern which can be very useful for market basket analysis. **Singh, R. et.al [2],** The paper specifies purpose of knowledge mining in educational dataset and summarized existing algorithm used in mining educational data and their gaps. Also it presents an algorithm that provides solution for the above mentioned negative points. It is an improved Apriori algorithm which reduces total time scanning number which is required for identifying frequent item set and association rules among education data using bottom up approach. The proposed algorithm also replaces the user-defined minimum threshold with standard deviation based on functional model as mentioned for minimum support value in advance which may lead to either very large and very less rule that have very negative impact on performance by the model. **Zhang, S. et.al (2015).[4],** In this paper a new mining tree problem is proposed whose goal is to find strictly create

modified way to create a tree using candidate key. This method is performed into two parts i.e. paired join & leg attachment. **Kumar, B. et.al (2010)[5],** Web Usage Mining is a data mining technique that finds out interesting usage patterns from Web data, for understanding and effective service needs of Web-based applications. Methods that are used: *Apriori Algorithm* is designed to operate on log files. In this various operations are performed in every modules. *FP-tree* is a frequent pattern tree consists of a root which is given "void" value in which there exists itemset prefix subtree in form of children of parent node, and candidate keys in tabular form. The main flaw when dealing with FP-growth algorithm is due to huge amount of dataset good candidate can't be produced. **Arora, J et.al (2013)[6],** In this paper, a review of four different association rule mining algorithms: Apriori, Apriori-Tid, Apriori-hybrid and tertius algorithms and their drawbacks which would be helpful to find new solution for the flaws that are present in these algorithms and also gives a comparitive approach for dealing with various association mining algorithms. **Rao, S. et.al(2012)[7],** This paper tells about FP-Growth algorithm which deals with mining of frequent itemsets by dividing the data into various fragments and then separately dealing with each one of them: Constructing précised structure of data called FP-tree which use 2 passes over the data-set. Extracting frequent itemsets from FP-tree traversal. This algorithm has various advantages as it is performs much faster than Researching-Tree and other association algorithms. The algorithm decrease amount of candidate itemed by introducing a compact form of the database in the form of the FP-tree. It stores important information thus helping for the efficient discovery of frequent item sets. But in spite of various benefits it can run short of memory, is costly to construct, takes time for construction. When deducing minimum support for large dataset we need to create tree for whole, which takes a very large amount of time for processing. **Dhanda, M et.al (2011)[8],** This paper deals with removing of flaws of ARM by using attributes like quantity and weight, weight attribute is used for estimation of amount of product that customer has bought, for calculating ratio of profit ratio and total profit value that an item is providing to the customer, the paper have used profit attribute. Here transaction database is used. Here ARM treating items in database with equality by taking in consideration the presence and absence of the items present in the transaction. It does not consider importance of items to business/user. The only problem with this method is that despite of doing optimization of memory it still needs much optimization.

III. PROPOSED ALGORITHM: MATRIX-OVER-APRIORI

A. *MATRIX-O-APRIORI:*

The proposed algorithm is totally based on matrix, so firstly we need to create a matrix based on the given dataset of problem. Then for calculating the support we need to perform various matrix operations like ANDing the rows for generating frequent candidate keys according to given minimum support. Our approach will not be scanning the dataset again and again. Further for getting final solution will add the entire individual AND solution into one.

B. *DESCRIPTION OF METHOD:*

1. Design operational matrix from the given dataset.

Preparing a matrix M rows m+1 items and c+1 coloumns from a given dataset having m items and c combinations. Futher simplification over that matrix is done by considering minimum support given. E.g when number of items is less than the support,it can't be the part of frequent itemset. Thus deletion of that row is done, similarly simplifying the matrix in subsequent steps.

2. Looking for L-Frequent Sets by performing AND

The largest frequent itemsets is a frequent itemsets which contains the most items than others.

a) Look up for the combinations having maximum count of items & mark the number as L. In case combination number is greater than value L, also we know the biggest frequent itemsets can't have L items, thus we will take combinations having L-1 items and then utilizing L to give L-1. Keeping on doing the same operation till combination having more minimum support is found.

b) Recursively keep on simplifying M by working with rows and column. Futher deleting those who are not having minimum support.

c) From b), there can be modification in column. Move to the step (a). In case (a) and (b) have not made any modifications in M then M will be renamed as NewM and move to step (c). In case M is void, there is no L-frequent itemsets, so we will move back on step (a) to deal with combinations having L-1 items.

d) Choosing combination "c" out of NewM, then taking out L items from c for to generate itemsets. In the next step "AND operation" is performed for counting itemsets support. In case support is larger than the required support provided in problem statement, the itemsets is an L-frequent itemsets. Otherwise it's not L-frequent itemset. After that re-selecting L items present in c and that too different from the previous and finding support for that. The same method is to be used till we stop getting different L items in c. Futher simplification of NewM is done in similarly as given in (b) by deleting the column having c. Continuing similarly with all other combinations present in NewM and and neglecting itemsets

previously taken. These method will be continued until NewM becomes void. In case there is no L-frequent itemsets, move to step (a) and for taking combinations having L-1 items.

3. Considering M look for all available frequent itemsets from 2 to L-1.

C. *PSEUDOCODE:*

Output will be frequent candidate itemset from the input dataset "A" and given minimum support.

1. M←Produce (A, SupportRequired);
   If M＝VOID
   END
2. BigL←Maximum (M, 0)
   While BigL>0，loop below given code
   NewM ← SimplifyM(M,BigL,SupportRequired)
   If NewM≠VOID
   QBigL←Find_Maximum(NewM, BigL)
   If QBigL≠VOID
   Break;
   BigL←Maximum (M, BigL)
   If QBigL＝VOID
   END
3. Q1←Detect_1(M)
   While L from 2 to BigL, execute
   M←SimplifyM(M, L, SupportRequired)
   CandL ← ProdCand(M, QBigL, ML-1)
   ML←ProdL(M, CandL, SupportRequired)
4. Solution←Q1∪Q2∪…∪QL∪QBigM∪M
5. [END]

D. *Why Matrix-O-Apriori is a better algorithm than existing algorithms?*

Apriori algorithm is most traditional and effective method for association rule mining. But it addition to so much positivity it also contains few unavoidable flaws. It gives rise to very large candidate's count for frequent items, as this algorithm do the scanning process frequently for frequent itemsets. So it is not that much efficient. Thus for dealing with the bottleneck in Apriori algorithm we have proposed a modification over apriori where matrix is used i.e. Matrix-O-Apriori. In this algorithm matrix is used in a very efficient manner highlighting the affairs that exists in the database upon which "AND operation" is applied to work upon elementary Matrix-operation which results in finding the biggest frequent itemsets. This algorithm doesn't scan the database again and again for optimization of affairs, and also greatly reduces the number of candidates of frequent itemsets.

Although in the second formula there is the time of finding the largest frequent itemsets, the P_K in the second formula is far less than the P_K in the first, and CK in the second also is far less than the CK in the first. The algorithm based on matrix don't do the scanning again and again, thus minimizing load of input-output units. Thus increase in the efficiency of proposed algorithm in terms of spatial and temporal complexity when compared to existing algorithms.

IV. EXPERIMENTAL RESULTS

A. *Datasets To Be Used:*

1. **Affair Database:** The database can be consisting of any type of affair regarding market, sequential, transaction and many more. In our example the it contains 8 affairs, Q＝{A1,A2,A3,A4,A5,A6,A7,A8}，the itemsets is I＝ {e,f,g,h,i,j,k,l},and the Support Required is 2. (Shown in table 1)

Table 1: Sample Data Set 1

Number of state	Itemsets
A1	h,i,k
A2	f,g,l
A3	h,g,e,j
A4	k,j,i,f
A5	k,i,h
A6	h,e,j,g
A7	k,i,j,f
A8	f,e,i,h,f

2. **Market-Basket Analysis** is a method which deals with if a customer buys certain products they are most likely to buy certain other products. Combination of goods bought by the customer is called **itemset**. This dataset finds out the logic between previous and next time when those goods were bought. This technique is used in determining the location and promotion of goods in a store. This method can be applied in various areas like:
    - Analysing purchase of credit cards.
    - Analysing calling patterns of telephone.
    - Determining fraud medical insurance claims.
    - Analysing purchase of telecom services.

    ❖ This is dataset (Table 2) from Market Basket . Support Required is 2 and Confidence is 1.5.

Table 2

TID	ITEMS
1	K, L
2	K, M,L,H
3	L, U, A,M
4	M,K,L,A
5	U,K,M,L

3. **Sequential Database:**

    Databases are a mixture of various sequence design. With the use of mining algorithm we need to determine the complete pattern set, with required support and that too by increasing scalability and efficiency. Also the method should avoid large number of scans and obeying various rules provided in the problem. In the following dataset we need to find frequent items with minimum support.

    ❖ A sequence database (Table 3) having Min Supp=0.5 and Min Confidence=0.5

Table 3

ID	SEQUENCES
M1	{q},{u,v}
M2	{p},{q},{u},{t}
M3	{p,s},{r},{q},{p,q,t,u}
M4	{p,q},{r},{u},{v},{t}

4. **Transaction Database:**

    Amount of operations done in DBMS with realiability and well-coordinated method which should be independent of other transactions. A transaction will represent when there is any change in database. The basic functionality of transactional database is:

    - Providing reliable amount of work due to which recovery from failure is correct, thus keeping consistency in database inspite of failure in system.

- Providing isolation in between programs those all are requiring database concurrently.
  - ❖ Transaction database (Table 4) table having Min Support=1 and Min Confidence=1

Table 4

TID	ITEMID
B1	b,a,e
B2	d,b
B3	c,b
B4	d,b,a
B5	c,a
B6	c,b
B7	c,a
B8	c,b,e,a
B8	c,b,a

B. *QUALITY MEASURES*

In our study, two classical algorithms are discussed i.e. FP Growth and Apriori Algorithm.

In **Apriori algorithm** extraction is done to find dataset association. Apriori algorithm doesn't provide the required efficiency as it is more time consuming algorithm in case of large dataset. It is not efficient in case of large dataset. The expending in time and space for frequent itemsets is too much. While in case of other methods can address the problem of frequent pattern mining with non-uniform minimum support threshold .So it is not useful in terms of efficiency, temporal and spatial complexities.

Another method discussed is **FP-Growth** which creates signatures of transactions on a tree structure to eliminate the need of database scans and outperforms compared to Apriori. Here there is no need to generate candidates. Horizontal and Vertical database layout is utilized to keep in Memory. The algorithm also has some drawbacks. Building of tree is very expensive and require very huge amount of storage capacity that is very difficult to get. Also its major flaw is that for creation of trees the dataset is scanned again and again, which is unavoidable for large datasets. Therefore to overcome the demerits of these algorithms we introduce a new algorithm named **MATRIX-O-APRIORI.**

MATRIX-O-APRIORI:

To solve loop hole of FP-Growth and Apriori , we have proposed a modified Apriori algorithm which is based on matrix. Different from existing algorithms, our method uses matrix and its methods. When compared with existing algorithms the proposed algorithm is very easy to perform and gives out clear result. Further AND Operation is used over matrix to produce frequent candidate item sets which don't scan the database time and again to lookup the affairs, and also greatly reduce the number of candidates of frequent item sets. It provides a better overall performance .It delete the useless transactions in the database for the purpose of reducing the size of database and reduced number of items. In brief, this algorithm provides a solution to the dynamic update problem. It handles both additions and deletions in increments and avoids a full database scan when the database is updated. Also, it avoids the construction of different matrices for mining frequent items with different support thresholds. Compared with the existing algorithms, this mining algorithm is more direct, integrated, convenient to maintain, especially fit for association rule mining in large-scale databases. Hence it greatly reduces I/O cost and it also doesn't generate irregular item set. So improved algorithm decreases time and space complexity and have higher efficiency as compared to our classical Apriori and FP Growth algorithm.

C. *Qualitative Result And Analysis*

In case of Matrix-O-Apriori with the use of matrix and its various operations it becomes simple, precice and thus much effective. When comparison is done with all previous algorithms, our proposed technique reduces the number of scans to much larger extent and too without taking much storage space with the use of huge matrix. Thus Matrix-O-Apriori makes the process of association rule mining clearer and subsequently providing an effective increase in data-mining technique (Figure 1).

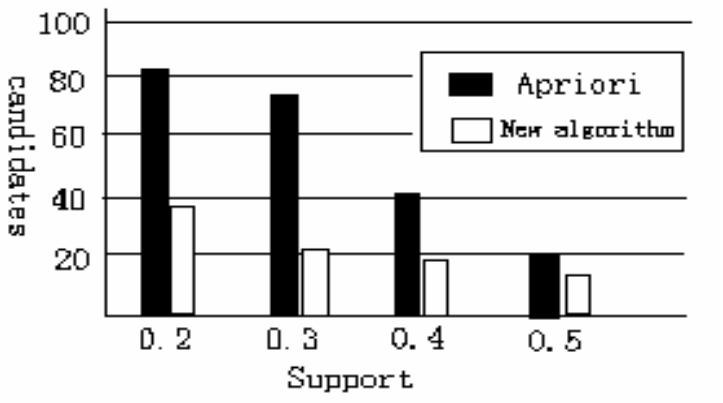

Fig 1: Comparison of number of candidates

D. *COMAPARATIVE RESULT*

Table 5: Comparative Results

PARAMETERS	APRIORI	FP GROWTH	MATRIX-O-APRIORI
Method	Basic Apriori using pruning technique.	Creating FP-Tree out of given dataset.	Use of matrix and AND operation to generate frequent sets.
Storage Space	Due to large number of candidates generated so require large memory space	Space required can be even more than the available memory	Very less amount of space used.
NO. of Scans	Multiple scans	Few scans	Scans only twice
Execution Time	High	Medium	Low
Efficiency	Low	Average	High
Spatial Complexity	High	Average	Low
Time Complexity	High	Low	Low

CONCLUSION

Basis of Matrix-O-Apriori algorithm is matrix which according to our Implementation process provides efficient spatial and temporal efficiency. This paper is a modification over various association rule mining. When given support is small the capacity of performance with this algorithm increase exponentially.Use of elementary matrix and AND Operation over it makes our proposed algorithm very scalable, precise, simple, clear and easy to implement. In addition to all the above benefits this method reduces memory space requirement to a much larger extent than previous ones.

ACKNOWLEDGEMENT

I am very thankful to Dr. Dharmveer Singh Rajput of Jaypee Institute Of Information Technology, Noida for his continuous guide and support with his knowledge and expertise on the basis of Association Rule Mining Techniques during my master's in Computer Science, without which this paper wouldn't be possible.

REFRENCES

[1] Jha, J., & Ragha, L. (2013). Educational Data Mining using Improved Apriori Algorithm. *International Journal of Information and Computation Technology*, *ISSN*, 0974-2239.
[2] Singh, R., & Chaudhary, S. Data Mining Approach Using Apriori Algorithm: The Review.
[3] Tseng, V. S., Wu, C. W., Fournier-Viger, P., & Yu, P. S. (2015). Efficient Algorithms for Mining the Concise and Lossless Representation of High Utility Itemsets. *Knowledge and Data Engineering, IEEE Transactions on*, *27*(3), 726-739.
[4] Zhang, S., Du, Z., & Wang, J. T. (2015). New Techniques for Mining Frequent Patterns in Unordered Trees.
[5] Kumar, B. S., & Rukmani, K. V. (2010). Implementation of web usage mining using APRIORI and FP growth algorithms. *Int. J. of Advanced Networking and Applications*, *1*(06), 400-404.
[6] Arora, J., Bhalla, N., & Rao, S. (2013). A Review on Association Rule Mining Algorithms. *IJIRCCE International Journal of Innovative Research in Computer and Communication Engineering*, *1*(5).
[7] Rao, S., & Gupta, P. (2012). Implementing Improved Algorithm Over APRIORI Data Mining Association Rule Algorithm 1.
[8] Dhanda, M., Guglani, S., & Gupta, G. (2011). Mining Efficient Association Rules Through Apriori Algorithm Using Attributes 1.
[9] Wang, C., Sun, W., Zhang, T., & Zhang, Y. (2009, August). Research on transaction-item association matrix mining algorithm in large-scale transaction database. In *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on* (Vol. 2, pp. 113-117). IEEE.