Analysis of Frequent Pattern Mining Using Association Rule Mining

[1]Hetal Khachane, [2]Hemali Savaliya, [3]Priyanka Raval

[1,2]PG Students, [3]Assistant Professor
Computer Engineering Department,
B.H.Gardi College Of Engineering & Technology
Rajkot, India

Abstract. **Association rule mining is very important technique of data mining. To finding frequent item set it is the most important task of association rule mining. There are numerous algorithms in frequent pattern mining. Apriori Algorithm is classical algorithm and most important algorithm of association rule mining. Apriori algorithm is inefficient due to multiple pass over the database, and if database is too large, then it will take too much time to scan the entire database. So it required more space and time. So this paper presents a survey on association rule mining using Apriori Algorithm.**

Index Terms: **Data mining, Association Rule, Apriori algorithm, support, confidence.**

I. INTRODUCTION

Data mining is interdisciplinary field of computer science. It is a computational process, which is discovering pattern into large dataset. Main goal of data mining is extract knowledge or information into huge amount of dataset. Association Rule are used to finding relation between any dataset. Association rules are if/then statements that help to expose relationships between unrelated data in a database, relational database or store other information. Association rules are used to find a relationship between the objects which are frequently used together. For example if the customer buys pizza

bread then he may also buy cheese.If the customer buys mobile then he may also buy memory card[1].

The problem is to generate all association rules that have support and confidence greater than the user-specified minimum support and minimum confidence.

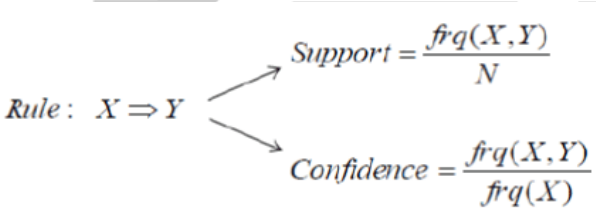

Fig 1: Association Rule[2]

Support(S):-Support(S) of an association rule is defined as the percentage/fraction of records that contain X∪Yto the total number of records in the database. Suppose the support of an item is 0.1%, it means only 0.1percent of the transaction contain purchasing of this item.

Confidence(C):- Confidence(C) of an association rule is defined as the percentage/fraction of the number of transactions that contain X∪Y to the total number of records that contain X. Confidence is a measure of strength of the association rules, suppose the confidence of the association rule X⇒Y is 80%, it means that 80% of the transactions that contain X also contain Y together[3].

II. TECHNIQUES OF ASSOCIATION RULE:

There are basically three types of association rule which are more used in recently:

- Apriori Algorithm
- FP-growth Algorithm
- Eclat Algorithm

Here below figure is various types of algorithm used in Association Rule:

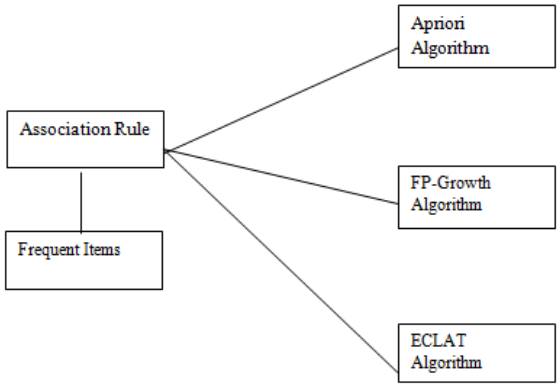

Fig 1: Various Types Of Algorithm Used In Association Rule

Frequent Item set:

Before seeing what frequent item is set First of all we can see what item set, support and support count.

Table 1: Transaction Table

TID	Items
1	Bread, Milk
2	Bread,Diaper,Beer,Eggs
3	Milk,Diaper,Beer,Coke
4	Bread,MilkDiaper,Beer
5	Bread,Milk,Diaper,Coke

Item set: Collection of one or more items. For example {bread, milk, butter}.**Support**: Fraction of Transactions that contain item set .For example s ({milk, Bread, Diaper)} =2. **Support count:** Frequency of occurrence of an item set. For example ({milk, Bread, Diaper}) =2.

Now **Frequent Item set** mean An Item set whose support is greater than or equal to a min_sup threshold[4].

Apriori Algorithm

Apriori algorithm to mine frequent itemset from transaction database for Boolean association rule. Here apriori algorithm is based on subset of frequent item set, so a subset of frequent item set must also be frequent item sets.i.e if {I1,I2} is a frequent itemset,then subset of {I1}& {I2} should be frequent itemset. The use of frequent item sets to generate association rule.

The algorithm uses a level-wise search, where k-item sets are used to explore (K+1) itemsets.In this algorithm, frequent subsets are extended one item at a time. This step is known as candidate generation process. Then groups of candidates are tested against the data. It identifies the frequent individual items in the database and extends then to larger and larger item sets as long as those item sets appear sufficiently often in the database.Apriori algorithm determine frequent item set that can be used to determine association rules which highlight general trends in the database.

Apriori algorithm takes advantage of the fact that any subset of a frequent itemset is also a frequent itemset.the algorithm can therefore, reduce the number of candidates being considered by only exploring the item sets whose support count is greater than the minimum support count. All infrequent item sets can be pruned if it has infrequent subset. So we build a candidate list of k-item sets and then extract a frequent list of k-item sets using the support count.

After that, we use the frequent list of k-item set in determining the candidate and frequent list of k+1 item sets so we use pruning to do that. We repeat until we have an empty candidate or frequent of k-itemsets, then we return the list of k-1 item sets[5][6][7].

Example of Apriori algorithm

Here we have six items for sale (Bread, Cheese, Eggs, Juice, Milk, and Yogurt) and we have Five Transaction given in Table, each transaction showing the purchases of one customer. We find association rules with 50% support. The transaction is given in Table:

Table 1: Transaction for Apriori

Transaction ID	Items
100	Bread,Cheese,Eggs,Juice
200	Bread,Cheese,Juice
300	Bread,Milk,Yogurt
400	Bread,Juice,Milk
500	Cheese,Juice,Milk

Scan Database for count of each Candidate,

Table 2 : Candidate 1-Itemset C1

Item set	Sup_Count
Bread	4
Cheese	3
Juice	4
Milk	3
Eggs	1
Yogurt	1

Now, Compare Candidate Support count with minimum support count.

Table 3 : Frequent item L_1

Itemset	Sup_Count
Bread	4
Cheese	3
Juice	4
Milk	3

Now Generate C_2 candidate from L_1,

Table 4: Candidate 2-Itemset C_2

Itemset	Sup_Count
(Bread,Cheese)	2
(Bread, Juice)	3
(Bread, Milk)	2
(Cheese, Juice)	3
(Cheese, Milk)	1
(Juice, Milk)	2

Compare candidate support count with minimum support,

Table 5: Frequent item L_2

Itemset	Sup_Count
(Bread,Juice)	3
(Cheese, Juice)	3

So Finally (Bread, Juice) and (Cheese, Juice) are sale together.

Advantage of Apriori:

- Depth first search reduces memory requirements.
- Usually faster than Apriori.
- No need to scan of (k+1) itemsets for K>=1.

Disadvantage of Apriori:

- The TID –sets can be quite long, hence expensive to manipulate.

A. FP-Growth

This is another important frequent pattern mining method, which generates frequent item set without candidate generation. It uses tree based structure. It works by generating a prefix-tree data structure known as FP-tree from two scans of the database. This method involves two phases. First phase need two database scans for generating the FPtree.But the second phase don't need any scan over database and it uses on FP-tree to generate frequent item set[8].

Example of FP-Growth:

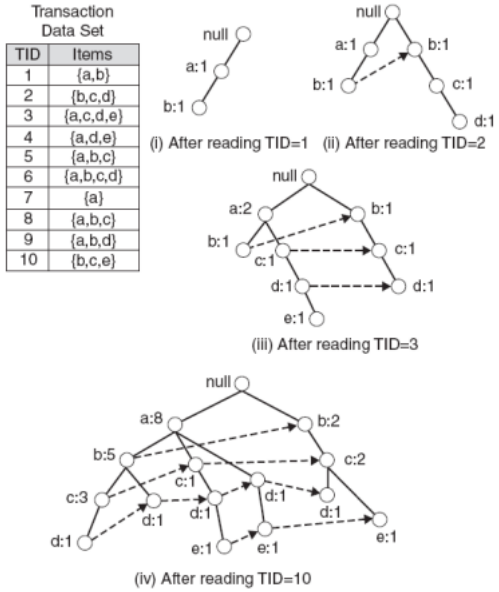

Fig2:- FP-Growth [9]

Advantage of FP-Growth:

- Only 2 pass over dataset.
- "Compresses "dataset.
- No candidate generation.
- Much faster than Apriori.

Disadvantage of FP-Growth:

- FP-Tree may not fit in Memory.
- FP-Tree is expensive to build.

ECLAT Algorithm

It is a depth first search based algorithm. Eclat algorithm uses a vertical database layout i.e. instead of explicitly listing all transactions; each item is stored together with its cover (also called tidlist) and uses the intersection based approach to compute the support of an item set [10].It requires less space than apriori if item sets are small in number .It is suitable for small datasets and requires less time for frequent pattern generation than apriori.

Example of ECLAT Algorithm:
Here we have Five items for sale (Bread, Butter, Jam, Coke, Milk) and we have Nine Transaction given in Table, Each transaction showing the purchases of one customer. We find ECLAT Algorithm with 50% support. The transactions are given in Table:

Table 6: Transaction for ECLAT

TID	Items
1	B1,B2,J1
2	B2,C1
3	B2,M1
4	B1,B2,C1
5	B1,M1
6	B2,M1
7	B1,M1
8	B1,B2,M1,J1
9	B1,B2,M1

Table 7: Frequent 1-Itemset

Item set	TID Set
B1	1,4,5,7,8,9
B2	1,2,3,4,6,8,9
M1	3,5,6,7,8,9
C1	2,4
J1	1,8

Table 8: Frequent 2-Itemset

Itemset	TID Set
{B1,B2}	1,4,8,9
{B1,M1}	5,7,8,9
{B1,C1}	4
{B1,J1}	1,8
{B2,M1}	3,6,8,9
{B2,C1}	2,4
{B2,J1}	1,8
{M1,J1}	8

Table 9: Frequent 3-Itemset

Itemset	Tid set
B1,B2,M1	8,9
B1,B2,J1	1,8

III. LITERATURE SURVEY

A. An improved Apriori Algorithm based on pruning Optimization and Transaction Reduction [11].

Chen et. al introduced pruned optimization strategy. With this strategy, the generation of frequent item sets is reduced and to compress the transaction of database, transaction reduction is used [8]. The value of the support selected is increasing which results in reducing number of frequent item sets generated. The comparison of proposed BE-Apriori algorithm has higher efficiency than the pure apriori algorithm.

B. A Prefixed-Itemset-Based Improvement for Apriori Algorithm [12].

Yu Shoujian proposed prefixed-item set-based data storage. With the help of prefixed-item set-based data storage, we managed to finish the connecting step and the pruning step of the Apriori algorithm much faster, besides They can store the candidate item sets with smaller storage space. Finally, they compare the efficiency of classical Apriori algorithm and improve Apriori algorithm on the aspect of support degree and the total number, and the experimental results on both aspects proved the feasibility of the prefixed-item set-based algorithm.

C. An Improved Apriori Algorithm For Association Rules[13]

Mohammed Al-Maolegi based on Apriori algorithm this paper indicates the limitation of the original Apriori algorithm of wasting time for scanning the whole database searching on the frequent item sets, and presents an improvement on Apriori by reducing that wasted time depending on scanning only some transactions. The time consumed to generate candidate support count in improved Apriori is less than the time consumed in the original Apriori; improved Apriori reduces the time consuming by 67.38%.

D. An Improved Apriori Algorithm[14]

Chang et.al proposed APRIORI-IMPROVE algorithm in which level L2 is directly generated from one scan over the database without generating candidate sets C1, L1 and C2. APRIORI-IMPROVE uses hash table and efficient horizontal data representation [9]. APRIORI-IMPROVE also optimized strategy of storage to save time & space The performance of APRIORI-IMPROVE is higher as compared to apriori and fpgrowth.

E. Research and improvement of Apriori Algorithm[15]

Jiaoling Du improved DC_Apriori algorithm, which restructured the storage structure of the database, improved connection of frequent item sets, the generation of k-frequent item sets is only need to join the frequent item sets with k-1-frequent item sets, greatly reduced the number of connections and it can directly get frequent item sets by only one pruning operation, effectively avoid the unnecessary invalid candidate sets, and greatly reduce the number of scanning the database and improve the efficiency of frequent item sets generation.

F. An Efficient Way To Find Frequent Pattern With Dynamic Programming Approach[16]

Dharmesh Bhalodiya improved Aprori-DP. In this research paper they describe the improved candidate 1-itemsets generation and candidate 2-itemsets generation from traditional technique. This algorithm utilizes the dynamic programming approach to facilitate fast candidate item set generation and searching.

TABLE OF COMPARISION I

Variations	Methodology	Problems to overcome	Results
A[6]	Pruning Optimization	Running time	Reduced running time but multiple scans still required
B[7]	prefixed-itemset-based data storage	efficiency	Improve efficiency and
C[8]	Pruning strategy	Running time	Reducing wasted time
D[9]	Hashing	Scanning and generation of candidate sets	Requires only one scan but Still large number of candidate sets generated.
E[10]	DC-Apriori	Frequently scanning database & generating large number of candidate itemset	Reduced the number of connection Directly get frequent itemset by only one pruning operation.
F[11]	Apriori-DP	Candidate itemset generation and scanning	facilitate fast candidate itemset Generation and searching.

IV. CONCLUSION

Association rule mining is an interesting topic of research in the field of data mining. We have presented a survey of most recent research work. However association rule mining is still in a stage of exploration and development. There are still some essential issues that need to be studied for identifying useful association rules. We hope that data mining researchers can solve these problems as soon as possible. Some problems for association rule mining are suggested below:

- To make frequent pattern mining an essential task in data mining, much research is needed.
- Most approaches are based on some strict assumptions. They should be generalized so that they can be more widely used.
- More efficient and scalable methods for Association Rule mining should be developed.
- Single scan and online mining methods should be developed.
- Database-independent measurements should be established. Deep-level association rule should be identified

REFERENCE

[1] Trupti A. Kumbhare Prof. Santosh V. Chobe,"An overview of Association rule Mining Algorithm" International Journal of Computer Science and Information Technologies,Vol. 5(1),pp. 927-930,2014.
[2] Suhani Nagpal," Improved Apriori Algorithm using logarithmic decoding and pruning", International Journal of Engineering Research and Applications, Vol. 2, Issue 3, pp.2569-2572, May-Jun 2012.
[3] Https://Www.Rroij.Com/Open-Access/A-Review-On-Association-Rulemining-Algorithms.Php?Aid=43382
[4] Http://Www.Cs.Kent.Edu/~Jin/Dm08/Fim.Pdf
[5] Jiawei Han And Micheline Kamber. Data Mining Concepts And Techniques. Second Edition, Morgan Kaufmann Publications, 2006.
[6] Agrawal, R. And Srikant, R. 1995." Mining Sequential Patterns", P. S. Yu And A. S. P. Chen, Eds.In:Ieee Computer Society Press, Taipei, Taiwan, 3{14}.
[7] Andrew Kusiak, Association Rules-The Apriori Algorithm[Online],Available: Http://Www.Engineering. Uiowa. Edu/~Comp /Public /Apriori.Pdf.
[8] Charanjeet Kaur," Association Rule Mining Using Apriori Algorithm: A Survey", International Journal Of Advanced Research In Computer Engineering & Technology (Ijarcet) Volume 2, Issue 6, June 2013.
[9] Https://Www.Google.Co.In/Search?Q=Fp-Growth&Client=Firefox B&Source=Lnms&Tbm=Isch&Sa=X&Ved=0ahukewjgqlqt9klrahvgsi8khtbkad Q_Auicigd&Biw=1366&Bih=657#Imgrc=Alc_Dgd3doqnum%3a
[10] Sumit Aggarwal, Vinay Singal," A Survey On Frequent Pattern Mining Algorithms", International Journal Of Engineering Research & Technology (Ijert) Ijertijert Issn: 2278-0181, Vol. 3 Issue 4, April – 2014.
[11] Zhuang Chen, Shibang Cai, Qiulin Song And Chonglai Zhu, "An Improved Apriori Algorithm Based On Pruning Optimization And Transaction Reduction,"Amisec 2011, 2ndieee International Conference, Pp1908-1911.
[12] Yu Shoujian, Zhou Yiyang," A Prefixed-Itemset-Based Improvement For Apriori Algorithm", College Of Computer Science And Technology, Donghua University, Shanghai, 2016.
[13] Mohammed Al-Maolegi, Bassam Arkok," An Improved Apriori Algorithm For Association Rules, International Journal On Natural Language Computing (Ijnlc) Vol. 3, No.1, February 2014.
[14] Rui Chang And Zhiyi Liu , " An Improved Apriori Algorithm," Iceoe 2011, Ieee International Conference, Vol. 1, Pp V1-476 -V1-478,2011..
[15] Jiaoling Du, Xiangli Zhang, Hongmei Zhang And Lei Chen," Research And Improvement Of Apriori Algorithm", Iceoe 2016, Ieee International Conference, Vol.1, Pp.117-121, 2016.
[16] Dharmshbhalodia, K.M. Patel ,Chhaya Patel, An Efficient Way To Find Frequent Pattern With Dynamic Programming Approach ,Nirma University International Conference On Engineering, Nuicone-2013, 28-30 November, 2013