International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Finding Frequent Pattern by Reducing Transctions Scan

Chandani Thakkar[1], Vinitkumar Gupta[2]

[1]Department of Computer Engineering, HGCE Vahelal, Ahmedabad, Gujarat, India

Abstract: ***Data mining is an important technology to help companies to extract hidden and considerable analytical information from their massive database. In data mining, we are finding frequent pattern using various techniques. Classical Apriori algorithm is a simple technique to find frequent item set from database. Classical Apriori algorithm scans database multiple times to find frequent item sets and generates large candidate sets, these are the disadvantages decrease efficiency and take more time. In this research paper, we describe improved apriori algorithm of reducing transactions scans. Proposed algorithm extracts efficient frequent pattern from transactional database by keeping records of transactions ids of each item. And also adds one temporary table after join operation to reduce size of candidate sets. This approach finds effective patterns with less time. This proposed algorithm can offer imperative suggestions to an organization for strengthening its business utility.***

Keywords: Data mining, Support,Association rule,Frequent pattern,Apriori Algorithm.

1. Introduction

As Information Technology is growing, databases created by the organizations are becoming huge. These organization sectors include marketing, telecommunications, manufacturing, banking, transportation etc. To extract valuable data, it is necessary to explore the database completely and efficiently. Data mining helps to identify valuable information in such huge databases. Data Mining is an analytic process designed to explore data in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. This is also known as "Knowledge Discovery" [1]. In other way, it is the extraction of hidden predictive information from large databases [13].

Many techniques have been developed in data mining amongst which primarily, Association rule mining is very important and best-known technique which results in association rules. Association rule mining is a technique for discovering unsuspected data dependencies [3] .Association rule mining is also known as "Frequent pattern mining". An association rule is having two important things support and confidence. It is an implication expression of the form X ->Y, where X and Y are disjoint item sets, i.e., $X \cap Y = \emptyset$. The strength of an association rule can be measured in terms of its support and confidence.

Support is the number of transactions in which the association rule holds. It is the percentage of transactions that demonstrate the rule. Support is defined as the occurrence of particular item among whole database [11].

Support (XY) = Support of ($X \cup Y$)/Total no. of transactions

Confidence is the conditional probability that, given X presents in transaction, Y will also be presents.

Confidence (XY) = Support count of ($X \cup Y$) / Support(Y)

Association rules describe the relations between the attribute values of any item set. Association rule can be found by support and confidence [11].The aim of association rule is to discover all association problems having support and confidence not less than the given value of threshold. If the support and confidence of item set of database is less than minimum support and confidence the n that item set is not frequent item set.

A. Apriori Algorithm

Apriori is very much basic algorithm of Association rule mining. It was initially proposed by R. Agrawal and R Srikant [1] in 1994 for mining frequent item sets. This algorithm uses prior knowledge of frequent item set properties that is why it is named as Apriori algorithm. It has to generate all rules that have support and confidence greater than a user-specified minimum support and minimum confidence respectively. Support is the percentage of transactions that demonstrate the rule. Apriori makes use of an iterative approach known as breath- first search[15], where k-1 item set are used to search k item sets. Classical apriori algorithm has two steps: 1) Join step 2) Prune step. The join step: Join operation is performed by itself to find new candidate sets [1]. The prune step: In this step Apriori first count the support value of each member and compare with predefined minimum support value. Which are not satisfy the min support value are been deleted.[2]

2. Previous Work

Historic data of algorithms for frequent pattern mining exists. Yubo Jia, Guanghu Xia have improved apriori algorithm by dividing database D into n parts which are not intersect with each other. Then find frequent pattern for each local part and combine them to make global frequent set. [7]

New improved algorithm for association rule mining finds all possible frequent patterns by each transaction wise. In this algorithm, database is scaned one time. It finds possible frequent items for each transaction.[8]

Bottom approach is also used to find frequent pattern. In this algorithm, first find largest frequent item from database using matrix. In matrix AND operation is performed between transactions and find largest frequent set. Then generates all possible subsets of them is called frequent item sets.[12]

Transaction reduction concept is used to improve apriori algorithm. In this algorithm, first calculate the size of transaction (SOT). First find single frequent item then check size of each transaction. If generates K-dimensional frequent patterns then delete those transactions which have size less then K. So, size of database is decreased by this approach.[14]

3. Enhanced Approach

In the proposed approach, try to reduce no. of transaction scans by this reduce time to find frequent pattern. In this proposed approach, store transactions ids for each item in which they are involved and count support value by finding common transactions of items. The approach given here is stated as RT-Apriori Algorithm (**R**educe **T**ransactions Scan). First calculates the support value for each item from database and also stores the transactions list for items in which they are involved. Then find support value for 2-dimensional items by counting common transactions of them. Now add one temporary table and find frequency of each item from L2. Compare frequency with minimum support value and delete those frequent items which containing this item. This approach reduces frequent items and running time.

A. *Algorithm*

Input: D= Database, minimum support
Begin
1) First scan database and find support value for each item.
2) Generate candidate set with support value and Transactions Ids of each item in which they are involved.
3) Compare support value of each item with minimum support value.
4) If support value is greater than minimum support value then go to step 5 else go to step 6.
5) Select frequent item and store in L1 with Transactions Id.
6) Reject frequent item.
7) Generate candidate set Ck-1 by 'join' step of Apriori algorithm.
8) Calculate support value for each item set by finding common transactions from L1.
9) Again compare support value with minimum support value. 10) If greater than then select frequent items with support value and Transactions Id and store into L2 otherwise reject frequent items
10) Add one temporary table in which find frequency of items which involved in the frequent item sets L2.
11) Compare frequency of an item if it is greater than 2 then go to step 13 otherwise go to step 14.
12) Select the frequent items from L2 in which that item is involved.
13) Delete the frequent items from L2 in which that item is involved.
14) Generate candidate set Ck by join operation on L2 and calculate support value for item sets by finding common transactions from L2.
15) Compare support value of item sets with minimum support value.
16) If greater than then select frequent items store into L3 otherwise reject frequent items.

End

B. Enlighnment of Algorithm

The very first step is to calculate support value for each item from the database. Also store the transaction ids in which item are involved. Table-1 shows transactional database. Here transactions with set of items. This table gives the idea about how many items customer purchase. Here min_sup=3.

Table 1: Transactional Database

T_id	Items
T1	A,B,C
T2	A,C,D,E
T3	B,C,D,F
T4	A,B,C,D
T5	A,B,C,F

Now generates candidate set C1 in which we calculate support value for each item and also store transactions id in which they are involved. Table-2 shows items with its support value and transactions id. This table gives idea about each item is how many times purchased by customers. This value is called support of item.

Generate L1 frequent item set from candidate set C1. Compare support value of each item with minimum support value. Accept those items which are greater than or equal to minimum support value, others are deleted. This table shows frequently purchased items. Here we store transactions id with items so can easily find support value for 2 and 3 dimensional frequent pattern.

Table 2: Candidate Set C1

Items	Support	T_id
A	4	T1,T2,T4,T5
B	4	T1,T3,T4,T5
C	5	T1,T2,T3,T4,T5
D	3	T2,T3,T4
E	1	T2
F	2	T3,T5

Table 3: Frequent Item set L1

Items	Support	T_id
A	4	T1,T2,T4,T5
B	4	T1,T3,T4,T5
C	5	T1,T2,T3,T4,T5
D	3	T2,T3,T4

Do join operation on L1 as original apriori algorithm and generates candidate set C2. Then find common transactions id of two items and count them for support value. In original apriori algorithm for to find support value should scan whole database so it takes more time. This table shows pair of frequent items.

Table 4: Candidate Set C2

Items	Support	T_id	
A,B	3	T1,T4,T5	
A,C	4	T1,T2,T4,T5	
A,D	2	T2,T4	Delete
B,C	4	T1,T3,T4,T5	
B,D	2	T3,T4	Delete
C,D	3	T2,T3,T4	

Compare support value with minimum support value. Here is min_sup=3. Select those items which are greater than or equal to minimum support value. This table shows pair of frequent items.

Table 5: Frequent Item Set L2

Items	Support	T_id
A,B	3	T1,T4,T5
A,C	4	T1,T2,T4,T5
B,C	4	T1,T3,T4,T5
C,D	3	T2,T3,T4

Add one temporary table after L2. In which we select unique items from L2 and find frequency of particular item. Frequency is the occurrence of item in L2. After this step we are generates 3-dimensional pattern so frequency of item should be greater than k-1 means 2. So select only those items which has frequency more than or equal to 2. By this step reduce the size of candidate set.

Table 6: Temporary Table

Items	Support	
A	2	
B	2	
C	3	
D	1	Delete

We compare frequency. As shown in table-4.6 item D is not greater than 2 so we delete that item and also delete those items from L2 which has D means deleted item. By this temporary table we can reduce size of candidate set and also save time. Table 7 shows frequent item set L2 generated after temporary table in which pair A,D is deleted.

Table 7: Frequent Item Set L2'

Items	Support	T_id
A,B	3	T1,T4,T5
A,C	4	T1,T2,T4,T5
B,C	4	T1,T3,T4,T5

Original apriori generates 4 frequent patterns as shown in table-8 and find support value for them. Whereas using L2' proposed algorithm generates only on pattern which is accepted as shown in table-9. Proposed algorithm generates small amount of frequent patterns.

Table 8: Frequent Set by Original Apriori

Items	Support	
A,B,C	3	
A,C,D	2	Delete
A,B,D	1	Delete
B,C,D	2	Delete

Table-9 Frequent Set by RT Apriori

Items	Support	T_id
I1,I2,I3	**3**	**T7,T8,T9**

Table-10 shows how many transactions are scanned by original apriori and improved algorithm. By improved algorithm improve efficiency of original apriori algorithm.

Table 10: No. of Transactions scanned

	Original Apriori	Improved Algorithm
1-itemset	30	30
2-itemset	30	18
3-itemset	20	3
Sum	80	51

4. Performance Evaluation

The algorithm is implemented on the large super market database. The proposed RT-algorithm trims down the execution time and time complexity. The testing of RT-algorithm and other traditional algorithm is done on the same database. The resultant chart in Fig-1,2 shows that proposed algorithm takes less time than apriori algorithm for different support value and different no. of transactions.

Table 11: Time comparison of two algorithms Under different Support value

	Time in seconds	
Support	*Original Apriori*	*Improved Algorithm (RT-Apriori)*
0.03	27	04
0.04	10	01
0.05	11	02
0.06	02	01
0.07	03	01

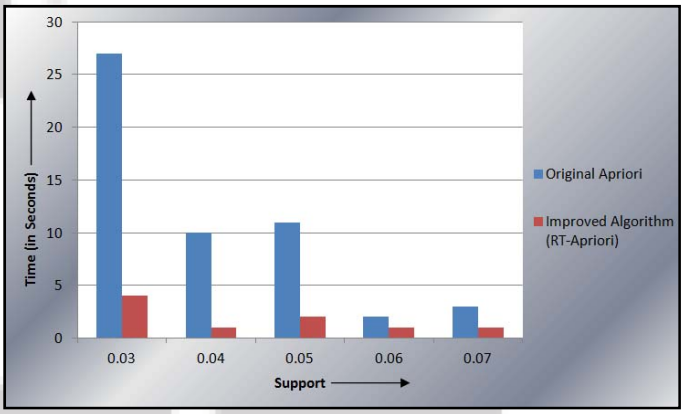

Figure 1: Comparison chart for Super market using two algorithms with different support value

Table 12: Time comparison of two algorithms for different no. of transactions

No. of transactions	*Time in seconds*	
	Original Apriori	*Improved Algorithm (RT-Apriori)*
300	07	02
500	09	03
800	10	02
1000	11	02

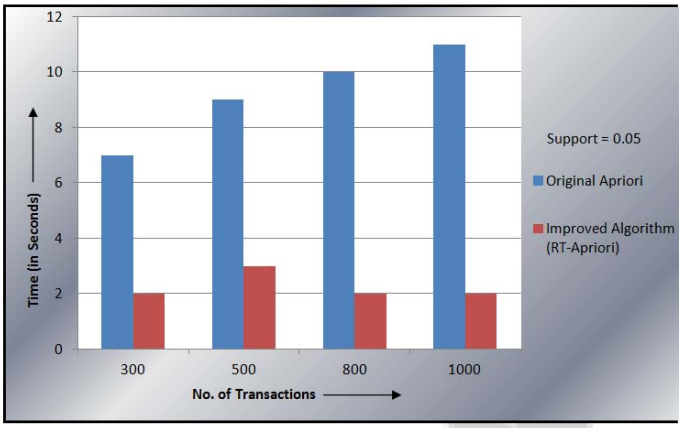

Figure 2: Comparison chart for Super market using two algorithms with different no. of transactions

5. Conclusion and Future Work

We reach to conclusion after analysis that proposed RT-Apriori algorithm takes less execution time than original apriori algorithm for finding frequent patterns. Proposed algorithm consumes less time than original algorithm for different support value and different no. of transactions. In this approach, we are storing transactions ids for each item and also add one temporary table for finding frequency of each item so it takes more memory than other algorithms. Future work is to reduce memory utilization by using some efficient method.

6. Acknowledgement

I would like to express my deep sense of gratitude to my guide, Asst Prof.Vinit Kumar Gupta for his valuable guidance and useful suggestions for this paper.

References

[1] Rakesh Agrawal,"Fast algorithm for Mining Association Rule", *Proceedings of the ACM SIGMOD International Conference Management of Data, Washington, 1993, pp.207-216.*

[2] Ekta Garg, Meenakshi Bansal, "A Survey On Improved Apriori Algorithm", *International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 7, July – 2013.*

[3] Kumar, "Association analysis: basic concepts and algorithms",http://www-users.cs.umn.edu ~kumar/dmbook/ch6.pdf

[4] Pranay bhandari, k. rajeswari, swati tonge, mahadev shindalkar "Improved Apriori Algorithms – A Survey", *International Journal of Advanced Computational Engineering and Networking, ISSN (PRINT): 2320-2106, Volume – 1, Issue – 2, 2013*

[5] Xiaojun Wang,"Study of Data Ming based on Apriori Algorithm", *2010 2nd International Conference on Software Technology and Engineering(ICSTE).*

[6] Zhuang Chen, Shibang Cai, Qiulin Song and Chonglai Zhu, "An Improved Apriori Algorithm Based on Pruning Optimization and Transaction Reduction", *2011 IEEE*

[7] Yubo Jia, Guanghu Xia, Hongdan Fan, Qian Zhang, Xu Li ,"Improved Apriori algorithm based on Association Analysis", *2012 Third International Conference on Networking and Distributed Computing*

[8] Girja Shankar, Latita Bargadiya," A New Improved Apriori Algorithm For Association Rules Mining", *International Journal of Engineering Research & Technology, Vol. 2 Issue 6, June – 2013*

[9] Sunil Kumar, "Improved Aprori Algorithm Based on bottom up approach using Probability and Matrix" , *IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 3, March 2012*

[10] Sheila A. Abaya," Association Rule Mining based on Apriori Algorithm in Minimizing Candidate Generation", *International Journal of Scientific & Engineering Research Volume 3, Issue 7, July-2012*

[11] Priyanka Asthan" Using Apriori Based Algorithm and Hash Based Methods", International Journal of Advanced Research in Computer ScieAsthana, Anju Singh,Diwakar Singh," *A Survey on Association Rule Mining and Software Engineering,Volume 3, Issue 7, July 2013*

[12] Shruti Aggarwal,"Comparative Study of Various Improved Versions of Apriori Algorithm", *International Journal of Engineering Trends and Technology (IJETT) - Volume4Issue4- April 2013*

[13] Tan, Steinbach, Kumar, "Introduction to Data mining", http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap6_basic_association_analysis.pdf

[14] Jaishree Singh, Hari Ram, Dr. J.S. Sodhi"Improving efficiency of Apriori Algorithm Using Transaction Reduction", *International Journal of Scientific and Research Publications, Volume 3, Issue 1, January 2013.*

[15] R.Divya , S.Vinod kumar ,"Survey on AIS,Apriori and FP-Tree algorithms",In: *International Journal of Computer Science and Management Research Vol 1 Issue 2 September 2012, ISSN 2278-733X*