ISSN:0975-9646

Finding Frequent Items Dynamically

Sunita Murjani[#], Indrajeet Rajput [*]

[#] *M.E.(CE),Gujarat Technological University, Ahmedabad,Gujarat, India*
**Asst. Professor, Department of Computer Engg., Gujarat Technological University, Ahmedabad,Gujarat India*

Abstract--Data mining place an important role in many of the applications like market–basket analysis, fraud detection etc. Frequent Pattern mining plays essential role in many of data mining task like association rules, causality, correlations, sequential pattern, multidimensional pattern etc. In Transactional Database, each transaction consists of items purchased by the customer. One of the basic market basket analysis algorithms is an Apriori, which generate all possible frequent patterns. In this research paper we describe the improved algorithm of dynamic Programming approach. This algorithm utilizes the frequent item set by combination of bottom up and top down approach to facilitate fast candidate item set generation and searching. This algorithm takes less than 2 scans of database for finding all frequent item sets. We have compared results with previous approach that optimize the database scans and eliminate duplicate candidate item set generation.

Keywords **- Association Rule Mining, Data Mining, Vertical data format, Dynamic Item set Counting Dynamic Programming, Frequent item sets.**

I . INTRODUCTION

Data mining means "The non trivial extraction of implicit, previously unknown, and potentially useful information from huge amount of data". Data mining also called as Knowledge Discovery in Databases (KDD). Many Business Enterprises accumulate large amount of data from their day to day operations. For Example Huge amount of customer purchase data daily occurs in any shopping mall. In such database each record represent transaction and attribute represent item purchase by customers.

Consider the example of super market. The transactional database of supermarket consist two attributes, transaction id, and items purchased by the customer. Each Transaction id is unique. The discovered patterns are set of items that are most frequent in database. For example, we want to find out how many of customer buys bread and jam together.

And how many customers buys bread and butter together. To find out such information apply market basket analysis and find out the pattern. Business person use this detail for identify the customer buying habits.

Association rule mining is used to find out relationship among items. Apriori [1] algorithm is used to find out association rules between set of items. Apriori algorithm requires minimum support, minimum threshold to find out frequent patterns from transactional database. There are two main functions in Apriori to find out association rules. First, it finds frequent item set based on minimum support count. After that minimum confidence is used to find out association rules between frequent items.

Dynamic programming is one of the techniques to design an efficient algorithm [3]. This technique works in bottom up manner and store the previous solutions in a table .so when the same problem encounter again no need to calculate again directly get from table which store the solutions. Many problems solve in an optimal way by dynamic programming approach. I.e. Matrix Chain Multiplications, Longest Common Sequence.

A. Apriori Algorithm

Let D be the database of any Supermarket. In Database D each row contains unique transaction identifier and items which are purchased by a customer. Let I be the item set {I1, I2, I3… In}. If an item set contain k-item then it called K-item set, and if K-item set satisfies the minimum support count and all of the subsets of K item set are frequent then it is called L_k frequent item set. In Apriori algorithm need to perform two basic steps are (a) Join, self join with previous frequent L_{k-1}itemset and create new candidate C_{k+1}itemset. (b) Prune, filter from the current candidate item set whose subset is not frequent in previous step. Below step explain the working of Apriori algorithm.

1. Assume that minimum support count and minimum confidence are given as min-sup and min-conf respectively.
2. Scan the entire database and find out candidate 1-item set C1.
3. Compare each item of C1 with min sup, if the item has support count less then min_supp than remove it.
4. Remaining 1-items in C1 which called L1.
5. Perform L1 Join L1, and create new C2, again scan the database and calculate occurrence of candidate 2-itemset appeared in database.
6. Apply pruning in C2 and generate L2.
7. Repeat Step 2 to 5 till we get generated set C_k is null.

II. PREVIOUS WORK

Numerous algorithms have been discussed for association rule mining. Rakesh Agarwal et al [2] introduced association rules for discovering regularity between products in large database. 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 [2]. Jaishree Singh introduced an attribute Size_Of_Transaction (SOT), containing number of items in individual transaction in database. The deletion process of transaction in database will made according to the value of K. Whatever the value of K, algorithm searches the same value for SOT in database. If value of K matches with value of SOT then delete only those transaction from database [3]. The Dynamic Programming approach [4] finds frequent -2 item set in one database scan.

The Dynamic Item set Counting algorithm [5] dynamically generate candidate item set as the algorithm proceed, it count varying length of item sets. As this way it requires less number of the database scans compare to Apriori. Finding Frequent Pattern with dynamic function [6] First generate the candidate pattern and prune by Apriori method. To count the support , instead of whole database for each pruned pattern we find longest common subsequence and length of transaction string of pattern's item and also stored new pattern and its transaction string so that next iteration we trace above string.

New Improved Apriori Algorithm for Association Rules Mining [7] uses the global power set to find frequent item set. In this scan database once and L1 is generated and from L1 global power set is generated. Global power set is all the possible combination of items which are in L1.And then scan database second time and if item set of global power set encounter then increment their count.

Statistical Approach for Data Mining to find the Frequent Item Sets [8] uses a formula max item support count minus current item set support count to found frequent pattern. Association Rule mining Based on matrix [9] uses matrix representation to represent items related to a transaction.

Efficient Association Rule Mining for XML Data [10] first convert XML Data to binary form and then AND operation is applied to find frequent item set. XOR operation is used to derive association rules.

III. PROPOSED ALGORITHM

We are using Dynamic Programming approach to find all possible frequent item set to improve the algorithm to find all frequent item set with dynamic programming approach [4]. In [4] Frequent item set with dynamic programming approach we can find only frequent-1 and frequent -2 item set. Our proposed algorithm overcomes this limitation. This approach requires less than 2 database scan finding all possible frequent item sets. This is the key feature to improve the performance. Next section shows the results and claimed that to generation of frequent item set in this technique is more efficient.

For example Consider transactional database that contains transactions carried out in super markets. In transactional database each transaction has unique identifier. Let the total no of items in supermarket are {I1, I2, I3, I4, I5}. Fig 1 shows the transactional database with 10 transactions.

FIG 1 SAMPLE DATABASE

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

Algorithm

Input: Transactional Database, No. Of different items

Method:

1. Scan each transaction in database and generate only one nth candidate item set based on size transaction.
   - 1.1 Merge Candidate nth item set based on size of transaction and if any item set is repeated increment the sup count of that item set.
   - 1.2 Compare sup count with min_supcount
   - 1.3 If supp_count is greater then min_supp then add it to the frequent n item set.
   - 1.4 From the results of previous step use subset property and generate all the frequent item set. (If item set-k size is frequent then all its subsets are frequent)
2. Using Vertical Data format find all the possible frequent item sets (For those transactions which are not considered as frequent in step 1)
   - 2.1 If sup count of item set is greater than 0 then check results of step1.If item set is present in the result of step1 then consider frequent also in step2. And if it is not present in the results of step1 then delete it.
3. Merge the results of step 1 and 2.
   - 3.1 If the same item set appears two times then write it once and add their support count in the final result.
4. All possible frequent Item sets.

Output: Frequent Item sets

Procedure contains two core parts. According to step1 of the Algorithm for above sample database based on the size of Transaction we are generating only one candidate-n item set for each of the transaction.

Size of transaction = no of items in transaction After applying the step 1 of the algorithm we get the candidate sets

shown in Figure 2.
Now apply step 1.1 of algorithm on the table shown in figure 1 we get table shown in Figure 3.
Now Apply Step 1.2 of algorithm:

FIG 2 CANDIDATE ITEM SET GENERATION

Tid	Candidate 2 Item set	Candidate 3 Item set
T1		I1,I2,I4
T2	I1,I2	
T3	I2,I4	
T4		I1,I2,I4
T5		I1,I3,I5
T6		I1,I3,I5
T7	I1,I2	
T8	I3,I5	
T9	I4,I5	
T10	I1,I3	

FIG 3 AFTER MERGING OF SAME ITEM SET

Tid	Candidate 2 Item set		Candidate 3 Item set	
	Itemset	Sup	Item set	Sup
T1,T4			I1,I2,I4	2
T2,T7	I1,I2	2		
T3	I2,I4	1		
T5,T6			I1,I3,I5	2
T8	I3,I5	1		
T9	I4,I5	1		
T10	I1,I3	1		

Min_sup=2
After comparing the item set of Figure 3 with min_sup we get Frequent Item sets in Step1
{I1, I2, I4 =2},{ I1, I3 , I5 = 2},{ I1,I2=2}
Now According to 1.5 step of algorithm find all the possible subsets because they are also frequent.
So we get

FIG 4 FREQUENT ITEM SETS

Frequent Item Set	Count
I1,I2,I4	2
I1,I4	2
I2,I4	2
I1,I3,I5	2
I1,I2	4(2+2)
I1,I3	2
I3,I5	2
I1,I5	2
I1	6(2+2)
I2	4
I3	2
I4	2
I5	2

Now According to Step 2 of algorithm now we Process only those transaction (T3, T8, T9, T10) which are not frequent in step1 using vertical Data format.
Generation of candidate 1 Item set

FIG 5 ITEM SET-1

Items	Transaction which contains item
I1	T10
I2	T3
I3	T8,T10
I4	T3,T9
I5	T8,T9

Now According to the Step 2.1 of the algorithm if the supp count is greater than 0 than check the results of step 1 .Since all items have supp greater than 0 and all are present in result of step 1. So C1=L1.
Now Generate candidate-2 item set

FIG 6 ITEM SET-2

Item set	Transaction which contains item
I1,I2	Null
I1,I3	T10
I1,I4	Null
I1,I5	Null
I2,I3	Null
I2,I4	T3
I2,I5	Null
I3,I4	Null
I3,I5	T8
I4,I5	T9

Now Again Apply 2.1 step of algorithm on table shown in Figure 6. So we get L2

FIG 7 FREQUENT ITEM SET-2

Item set	Count
I1,I3	1
I2,I4	1
I3,I5	1

Now Generate candidate 3 item set.

FIG 8 ITEM SET-3

Item set	Count
I1,I3,I5	0

Since Item set 3 have 0 supp count.
So we will stop here.
Now According to Step 3 of algorithm add the results of step 1 and step2.Finally we get Frequent Items Shown in Figure 9

FIG 9 FREQUENT ITEMS

Item set	Supp_count
I1	7
I2	5
I3	4
I4	4
I5	4
I1,I2	4
I1,I3	3
I3,I5	3
I1,I5	2
I1,I4	2
I2,I4	3
I1,I3,I5	2
I1,I2,I4	2

In [4] Frequent item set with dynamic programming approach we can find only frequent-1 and frequent -2 item set. In our algorithm it takes less than 2 scan of database to find all the frequent Items. Step 1 of the proposed algorithm works in bottom up manner to find frequent Item set and Step 2 of the algorithm works in top down manner to find frequent item sets. At last according to step 3 of algorithm by combining the results of step 1 and 2 we get all frequent item sets. In this approach it clearly examine that it has advantage over Dynamic programming approach [4].

IV. Experiments And Results

We have used Intel Pentium 1 GHz or higher with 2 GB RAM or more 3 GHz system for experimental work. We have created dummy database of 1000 records, 1500 records, 2000 records, 2500 records with 50 items. Experiment work of traditional Apriori and proposed Dynamic Method, we refers as Dynamic Method.

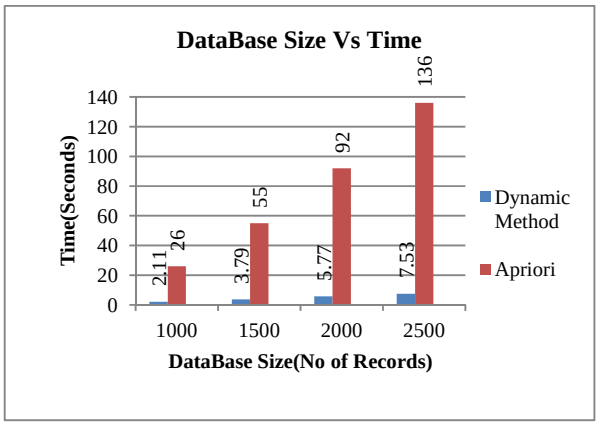
1. *Time Comparison*

Fig 10 represents bar chart of time requirement of individual algorithms. X-axis represents database size with the interval of 500 transactions. And Y-axis represents vertically with time. We can see in the Fig 10 at all the levels Dynamic Method takes less time compare to Apriori Method.

2. *Memory comparison*

Memory Requirement is of an algorithm is total space taken by the algorithm with respect to the input size. Memory

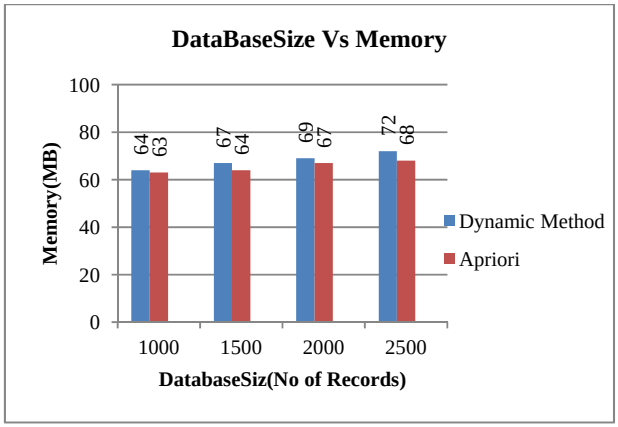
Fig 10 Time Comparison



Requirement includes both Auxiliary space and space used by input. Auxiliary Space is the extra space or temporary space used by an algorithm.

As it is clear from Figure 11, the memory consumption for the Apriori algorithm is the less at all level compare to our proposed algorithm because in proposed algorithm first we generate only one candidate- n item set for all transactions and then perform computation for support count. After getting results from previous step we are saving results into the memory and the item set which are not frequent their transaction id will be saved and those transactions are processed using vertical data format. At every level of finding frequent item set we are using result of first step.

Figure 11 Memory COMPARISONS



3. *Reliability*

A pattern is reliable if the relationship described by the pattern occurs in a high percentage of applicable cases. For example, a classification rule is reliable if its predictions are highly accurate, and an association rule is reliable if it has high confidence.

In our Experiment after finding frequent item set we are finding interesting association rules based on the minimum confidence.

Confidence= A→B= P(B/A)=P(AUB)/P(A)

3.1 *Lift/Interest:*

Sometimes it may be possible the rules have high confidence but it may be not interesting. To find out such rule lift is used. In association rule A→B The occurrence of item set A is depended on B or both are Independent we can know through lift.

Lift= confidence (A→B)/P(B)

If lift<1 then item set A and B are negatively correlated.

If lift>1 then item set A and B are positively correlated.

If lift==1 then item set A and B are independent.

Improvement in Dynamic Method comes from two major factors. First is reducing one database scan and find all the possible frequent item sets. As we know in step1 of the proposed algorithm it performs scanning of all database records

Consider k = no of frequent transaction of step

So consider time for step1 is n. In step 2 we have to scan only those transactions which are not frequent in step 1.

I = n-k

So Total Scanning time for our proposed algorithm is O(ni).

V. CONCLUSIONS

We have proposed changes in dynamic Programming approach with use of efficient memory and combination of bottom up and top down methodology. Step1 of algorithm works in bottom up manner and step 2 works in top down manner. The effective improvement in proposed by finding all the possible frequent item set in O(ni) time. Proposed algorithm takes very less time to compute results. We also presented comparative analysis of traditional approach (Apriori) and our approach with reference to CPU overhead with time requirement to complete the task and memory requirement, finally observed the proposed algorithm required very less time than Apriori algorithm.

ACKNOWLEDGEMENT

We are very thankful to Indra Jeet Rajput , Asst Prof in Computer Department, Hasmukh Goswami College of Engineering, Vahelal, Gujarat, for proving base resource for experimental work and giving guidance when required while research work.

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 Bansa l, " A Survey On Improved Apriori Algorithm", *International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 7, July – 2013.*

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

[4] Dharmesh Bhalodiya, 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*

[5] SergyBrin , Rajeev Motwani , Jefrey D Ulman ,Shalom Tsur "Dynamic Item set Counting and Implication rules for Market BasketAnalysis" *ACM SIGMOD conference. Management of Data,May 1997.*

[6] Sunil Joshi, Dr. R. S. Jadon , Dr. R. C. Jain " An Implementation of Frequent Pattern Mining Algorithm using Dynamic Function" *International Journal of Computer Applications (0975 – 8887) Volume 9– No.9, November 2010*

[7]Girja Shankar,LatitaBargadiya, "A New Improved Apriori Algorithm For Association Rules Mining" , *International Journal of Engineering Research & Technology (IJERT) ISSN No- 2278-0181 Vol. 2 Issue 6, June –2013*

[8] N.Venkateshwarlu "Statistical Approach for Data Mining to find the Frequent Item Sets "*International Journal of Latest Trends in Engineering and Technology (IJLTET) ,Vol 2 issue 2 March 2013ISSN No -2278-621X.*

[9] X. Luo and W. Wang, "Improved Algorithms Research for Association Rule Based on Matrix," *2010 International Conference on Intelligent Computing and Cognitive Informatics*, pp. 415–419, Jun. 2010.

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

[11] Myint Khaing, NilarThein " An Efficient Association Rule Mining For XML Data" *SICE-ICASE International Joint Conference 2006 ISSN No-89-950038-5-5.*