ENHANCING THE APRIORI ALGORITHM USING SET THEORY IN ASSOCIATION RULE MINING

Rahul H. Sathawane[1], Alok Chauhan[2] , Santosh Kumar Sahu3 , Akhil Anjikar[4] , Ritesh Shrivastava[5]

[1,2,3,4] *Information Technology, RGCER/RTMNU, (India)*

[5]*Computer Science and Engineering, ACOE/RTMNU, (India)*

ABSTRACT

The Apriori Algorithm is an influential algorithm for mining frequent item sets for Boolean association rules. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search(BFS) and a Hash tree structure to count candidate item sets efficiently.

Efficient algorithms to mine frequent patterns are crucial in data mining. Since the Apriori algorithm was proposed to generate the frequent item sets, there have been several methods proposed to improve its performance. But they do not satisfy the time constraint. However, most still adopt its candidate set generation-and-test approach. In addition, many methods do not generate all frequent patterns, making them inadequate to derive association rules. The Enhance apriori algorithm has proposed in this paper requires less time in comparison to apriori algorithm. So finally time is reducing with this different approach.

Keywords : Apriori Property , BFS , Frequent Itemsets ,Generate and test , Join Operation

1. INTRODUCTION

Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. An example of an association rule would be "If a customer buys a dozen eggs, he is 80% likely to also purchase milk. An association rule has two parts, an antecedent (if) and a consequent (then). An antecedent is an item found in the data. A consequent is an item that is found in combination with the antecedent. Association rules are created by analyzing data for frequent if/then patterns and using the criteria support and confidence to identify the most important relationships. Support is an indication of how frequently the items appear in the database. Confidence indicates the number of times the if/then statements have been found to be true.

In data mining**,** association rules are useful for analyzing and predicting customer behavior. They play an important part in shopping basket data analysis, product clustering, and catalog design and store layout. Programmers use association rules to build programs capable of machine learning. Machine learning is a type of artificial intelligence (AI) that seeks to build programs with the ability to become more efficient without being explicitly programmed.

Association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness. Data mining uses information from past data to analyze the outcome of a particular problem or situation that may arise. Data mining works to analyze data stored in data warehouses that are used to store that data that is being analyzed. That particular data may come from all parts of business, from the production to the management. Managers also use data mining to decide upon marketing strategies for their product. They can use data to compare and contrast among competitors.

Association rule mining is used in many applications

For example:

- Point of sales systems to improve Sales.
- Analyze Web usage patterns.
- Banking and Finance.
- Stock Forecasting.
- Adverse drug reaction detection.
- Market Basket Analysis.
- Medicare & Health Care**.**

II APRIORI ALGORITHM

Apriori is a classic algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database. This has applications in domains such as market basket analysis.

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time a step known as *candidate generation*, and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation).

III METHODOLOGY

- The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules.

Key Concepts:

- Frequent Itemsets: The sets of item which has minimum support (denoted by L_i for ith-Itemset).
- Apriori Property: Any subset of frequent itemset must be frequent.
- Join Operation: To find L_k , a set of candidate k-itemsets is generated by joining L_k-1 with itself.

3.1 The Apriori Algorithm: Pseudo code

*L*1 = find frequent 1-itemsets(D);

for (*k* = 2;*Lk-1* ≠ Φ;*k*++)

{

Ck = Apriori gen(*Lk*-1);

for each transaction *t* Є *D*

{

// scan *D* for counts

Ct = subset(*Ck*, *t*);

// get the subsets of *t* that are candidates

for each candidate *c* Є *Ct*

c.count++;

}

Lk = {*c* Є *Ck* | *c.count* ≥ *min sup}*

}

return *L* = U*kLk*;

3.2 Procedure Apriori gen(*Lk-1*:frequent (*k-1*)-itemsets)

for each itemset *l*1 Є *Lk*-1

for each itemset *l*2 Є *Lk*-1

if (*l*1[1] = *l*2[1])^(*l*1[2] = *l*2[2])^:::^(*l*1[*k*-2] = *l*2[*k*-2])^(*l*1[*k*-1] < *l*2[*k-1*])

then

{

c = *l*1 x *l*2; // join step: generate candidates

If has infrequent subset(*c*, *Lk*-1) then

delete *c*; // prune step: remove unfruitful candidate

else add *c* to *Ck*;

}

return *Ck*;

3.3 Procedure has infrequent subset

(*c*: candidate *k*-itemset;*Lk*-1: frequent (*k*-1)-itemsets); // use prior knowledge

for each (*k*-1)-subset *s* of *c*

if *s* ¢ *Lk*-1 then

return TRUE;

return FALSE;

3.4 Market Basket Analysis Example

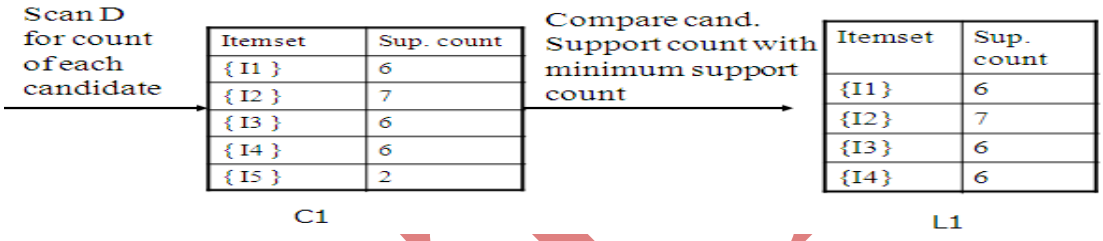
Market basket analysis can be used in deciding the location and promotion of goods inside a store. If, as has been observed, purchasers of Barbie dolls have are more likely to buy candy, then high-margin candy can be placed near to the Barbie doll display. Customers who would have bought candy with their Barbie dolls *had they thought of it* will now be suitably tempted. But this is only the first level of analysis. Differential market basket analysis can find interesting results and can also eliminate the problem of a potentially high volume of trivial results. In differential analysis, we compare results between different stores, between customers in different demographic groups, between different days of the week, different seasons of the year, etc.

Example:

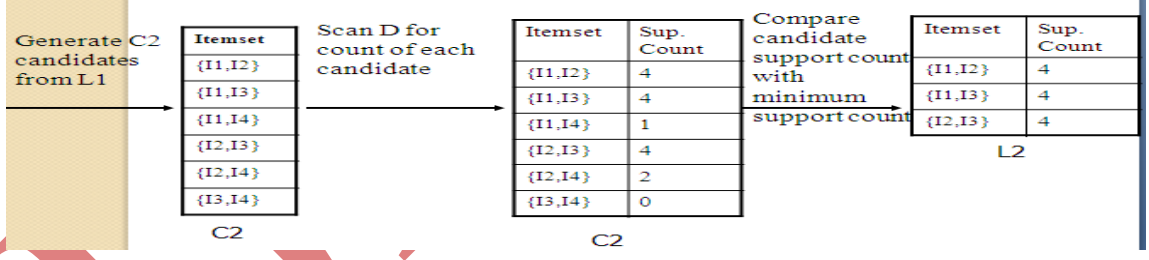
Transaction	List of Item Set
1	I1,I2,I5
2	I2,I4
3	I2,I3
4	I1,I2,I4
5	I1,I3
6	I2,I3

7	I1,I3
8	I1,I2,I3,I5
9	I1,I2,I3
10	I4
11	I4
12	I4
13	I4

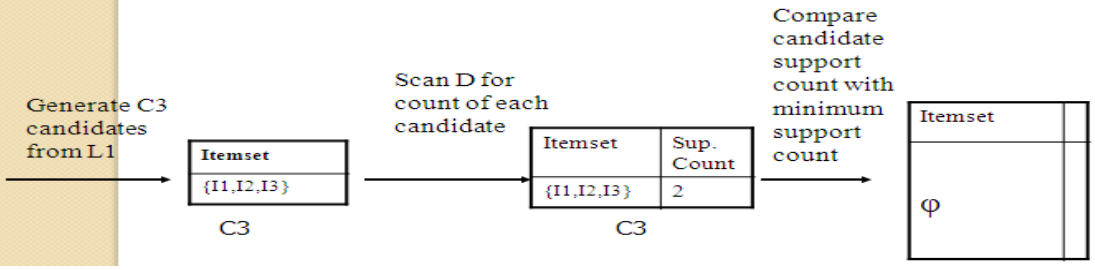
Step 1: Generating 1-itemset Frequent Pattern



Step 2: Generating 2-itemset Frequent Pattern



Step 3: Generating 3-itemset Frequent Pattern



•The algorithm uses L2 *Join* L2 to generate a candidate set of 3-itemsets, C3. Although the join results in {I1, I2, I3}, this itemset L3 is not frequent.
• Thus, L3 = φ , and algorithm terminates, having found all of the frequent items. This completes Apriori Algorithm.

3.5 Enhancement in apriori Algorithm

The proposed algorithm shows the enhancement in apriori algorithm which reduces the time for generating the frequent item. The main problem in the apriori is that it takes the number of passes which is equal to the max length of frequent item set. In our approach we are using the intersection method which will reduce the time.

3.5.1 Enhance Apriori algorithm

Initialize: K: = 1, C1 = all the 1- item sets;

read the database to count the support of

C1 to determine L1.

L1 := {frequent 1- item sets};

k:=2; //k represents the pass number//

while (Lk-1 ≠ ∅) do

begin

Ck: = gen_candidate_itemsets with the given Lk-1

Prune (Ck)

for all candidates in Ck do

count the number of transactions by using intersect method that are common in each item ∈ Ck

Lk := All candidates in Ck with minimum support ;

k := k + 1;

end

Answer := ∪k Lk ;

Here we are presenting a different approach in Apriori algorithm to count the support of candidate item set. Basically this approach is more appropriate for vertical data layout, since Apriori basically works on horizontal data layout. In this new approach, we use the set theory concept of intersection. In Classical Apriori algorithm, to count the support of candidate set each record is scanned one by one and check the existence of each candidate, if

candidate exists then we increase the support by one. This process takes a lot of time, requires iterative scan of whole database for each candidate set, which is equal to the max length of candidate item set. In modified approach, to calculate the support we count the common transaction that contains in each element's of candidate set, by using the intersect query. This approach requires very less time as compared to classical Apriori.

IV RESULTS AND ANALYSIS

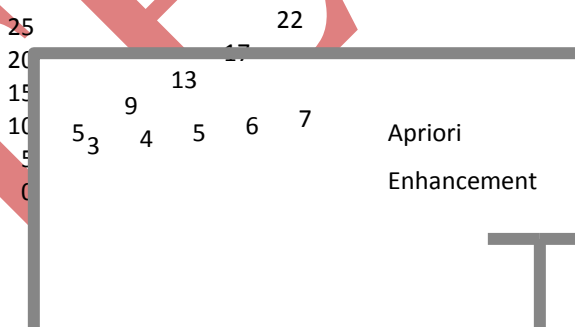
4.1 Results of Apriori and Enhance Algorithm:

We have taken support count 10% and number of item 10 with different number of transaction.

Table for time comparison

Transactions	Apriori	Enhancement in Apriori
2000	5 Seconds	3 Seconds
4000	9 Seconds	4 Seconds
6000	13 Seconds	5 Seconds
8000	17 Seconds	6 Seconds
10000	22 Seconds	7 Seconds

Graph for time comparison with respect to number of transactions



V SUMMARY

Association rules mining, introduced by Agrawal, has been traditionally applied in databases of sales transactions (referred to as market basket data). Different algorithms have been proposed for finding frequent item sets. The **Apriori Algorithm** is a well-known approach which is proposed by Agrawal & Srikant (1994). It is an iterative approach and there are two steps in each iteration. The first step generates a set of candidate item sets. Then, in the second step we count the occurrence of each candidate set in database and prune all disqualified candidates (i.e. all infrequent item sets). Apriori uses two pruning technique, first on the bases of support count (should be greater than user specified support threshold) and second for an item set to be frequent , all its subset should be in last frequent item set.

5.1 Enhancement in apriori Algorithm

The proposed algorithm shows the enhancement in apriori algorithm which reduces the time for generating the frequent item. The main problem in the apriori is that it takes the number of passes which is equal to the max length of frequent item set. In our approach we are using the intersection method which will reduce the time.

VI CONCLUSION

Throughout the last decade, a lot of people have implemented and compared several algorithms that try to solve the frequent item set mining problem as efficiently as possible. For example, a very often used implementation of the Apriori algorithm is that by S. Prakash R. M. S. Parvathi, An Enhanced Scaling Apriori for Association Rule Mining Efficiency. Nevertheless, when we compared his implementation [5] with ours, the performance of both algorithms showed immense differences. We can conclude that in this new approach, we have the key ideas of reducing time. As we have proved above how the Enhance apriori take less time than that of classical algorithms. That is really going to be fruitful in saving the time in case of large database. This key idea is surely going to open a new gateway for the upcoming researcher to work in the field of the data mining.

VII FUTURE WORK

- Test the Apriori algorithm in large data set.
- Enhancement of Apriori algorithm by using cardinality count.
- Optimized by candidate generation step.
- Saves computation time.

REFERENCES

[1]. Agrawal, R., Imielinski, T., and Swami, A. N. 1993. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 207-216.

[2]. Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules. In Proc. 20th Int. Conf. Very Large Data Bases, 487-499.

[3]. Agarwal, R. Agarwal, C. and Prasad V., A tree projection algorithm for generation of frequent item sets. In J. Parallel and Distributed Computing, 2000.

[4]. J. Han, J. Pei, and Y. Yin,"Mining Frequent Patterns without Candidate Generation," Proceedings of ACM SIGMOD International Conference on Management of Data, ACM Press, Dallas, Texas, pp. 1-12, May 2000.

[5]. Vaibhav Kant Singh and Vinay Kumar Singh "Minimizing Space Time Complexity by RSTDB a new method for Frequent Pattern Mining" To be appeared in proceeding of the First International Conference on Intelligent Human Computer Interaction ,Allahabad,2009.

[6]. M. S. Chen, J. Han, and P. S. Yu. Data mining: An overview from a database Perspective. IEEE Trans. Knowledge and Data Engineering, 8:866-883, 1996.

[7]. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.

[8]. W. J. Frawley, G. Piatetsky-Shapiro and C. J. Matheus, Knowledge Discovery in Databases: An Overview. In G. Piatetsky-Shapiro et al. (eds.), Knowledge Discovery in Databases. AAAI/MIT Press, 1991.

[9]. Osmar R. Zaïane, 1999 CMPUT690 Principles of Knowledge Discovery in Databases. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2009.