International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Analysis of Knowledge Set Discovery in Mining Items with Enhanced Apriori Association Algorithm

Gajula Bharathi[1], Gunna Kishore[2]

[1, 2]Computer Science and Engineering, Rise Group of Institutions, Ongole, India

Abstract: ***Frequent item generation is a key approach in association rule mining. The Data mining is the process of generating frequent itemsets that satisfy minimum support. 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 the time is reducing.***

Keywords: Apriori, Item set, Frequent Item set, Support count, threshold, Confidence

1. Introduction

Data Mining is a promising and flourishing frontier in database systems and new database applications. Data mining [2] is the process of finding interesting trends or patterns in large data sets to guide decision about future activities. It is the analysis of dataset to find unsuspected relationship and to summarize the data in new ways that are both understandable and useful. Progress in digital data acquisition and storage technology has resulted in growth of huge database. Data is often noisy and incomplete, it is likely that many interesting patterns will be missed and reliability of detected patterns will be low.

So knowledge Discovery in databases (KDD) and Data Mining (DM) helps to extract useful information from raw data. Frequent patterns are ones that occur at least a user-given number of times (minimum support) in the dataset. They allow us to perform essential tasks such as discovering association relationships among items, correlation, sequential pattern mining, and much more.

Association rules mining, introduced by Agrawal, has been traditionally applied in databases of sales transactions (referred to as market basket data). A transaction T is a set of items and contains an item set I if $I \subseteq T$ If I has k members, then I is called a k_itemset. An association rule is an implication X→Y where X and Y are itemsets with no items in common i.e. X∩Y =∅. The intuitive meaning of such a rule is that the transactions (or tuples) that contain X also contain Y. The rule X→Y holds with confidence c if c% of transactions that contain X also contain Y. The rule $X \rightarrow Y$ has a support s if s% of the transactions in the database contains X ∪Y. Given a database, the problem of mining association rules is to generate all rules that have support and confidence greater than the user-specified minimum thresholds, min-Support and min-Confidence.

Association rules processing is usually broken down into two sub problems:
1. Finding all frequent itemsets (whose supports are greater than the min-Support), also called covering or large itemsets in the literature.
2. Generating association rules derived from the frequent itemsets.

The association rules technique has also been applied to tabular data sets. An example of an association rule in tabular data is as follows :(A1= 2) and (A2 = 3) and (A4 = 5) →A6 = 1 support = 10%; Confidence= 60%
Where A1,….. ,A6 are attributes.

1.1 KDD Steps used in Generating Frequent Item set

The knowledge discovery [7] in databases follows certain steps that are given below:
1) Domain Knowledge
2) Examining the entire raw dataset identifies finding target dataset- the target subset of data and the attributes of interest 3. Data cleaning, data reduction, and data transformation
3) Choosing data mining task and algorithms
4) Knowledge discovery

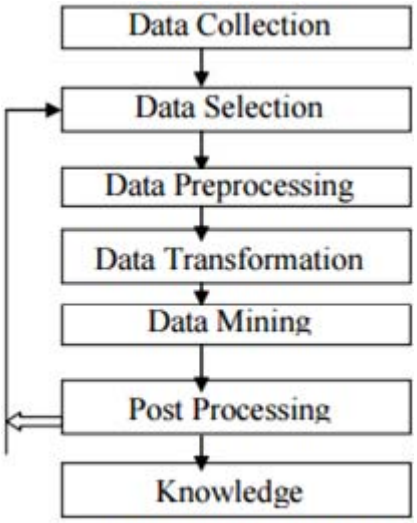

Figure1.1: Steps of KDD

In applying various data mining techniques to the data source of Interest, different knowledge comes out as the mining result. This knowledge [9] is evaluated by certain rules, such as the domain knowledge or concepts. After the evaluation as shown in fig 1.1, if the result does not satisfy the

requirements then we have to redo some processes until getting the correct results. The results can be projected as raw data, tables, and decision trees. The main objective of the KDD process is to make the data mining results easier to be used and more understandable.

2. Apriori Algorithm for Generating Frequent Itemsets

Different algorithms have been proposed for finding frequent item sets. The **Apriori Algorithm** is a well-known approach which is proposed by Agrawal & Srikant [1] (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.

The iterations begin with size 2 item sets and the size is incremented after each iteration. The algorithm is based on the **closure property [2]** of frequent item sets: if a set of items is frequent, then all its proper subsets are also frequent.

Apriori Algorithm

Initialize: k := 1, C_1 = all the 1- item sets;
read the database to count the support of C_1 to determine L_1.
L_1 := {frequent 1- item sets};
k:=2; //k represents the pass number//
while ($L_{k-1} \neq \emptyset$) do
begin
C_k:= gen_candidate_itemsets with the given Lk-1
prune(C_k)
for all transactions t ∈ T do
increment the count of all candidates in C_k that are contained in t;
L_k:= All candidates in Ck with minimum support ;
k := k + 1;
end
Answer := $u_k \cup L_k$;

The first weakness of this algorithm is the generation of a large number of candidate item sets. The second problem is the number of database passes which is equal to the max length of frequent item set.

3. 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.

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 ($L_{k-1} \neq \emptyset$) do
begin C_k): = gen_candidate_itemsets with the given L_{k-1} Prune (C_k))
for all candidates in Ck do
count the number of transactions by using intersect method that are common in each item ∈ C_k
L_k:= All candidates in C_k with minimum support ;
k := k + 1;
end
Answer := ∪k Lk ;

Although it is a first algorithm proposed in this field of frequent pattern mining [3] , but with the time a number of modified algorithm were designed to improve the efficiency of time, memory management and remove the complexity of process. 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 of SQL. This approach requires very less time as compared to classical Apriori.

4. Comparison between Apriori and Enhance Apriori

The Application is developed using Visual Basic 6.0 and oracles 9i.we have used the synthetic database. We have not applied this approach to any specific domain. We suggest the general approach which will be apply to any domain.

4.1 Time Comparison in Apriori and Enhance Apriori Algorithm Increasing Number of Transaction

For the comparative study of Apriori and Enhance Apriori algorithm we have taken a database of 1000 transaction of 10 items. In this analytical process we considered 200 transactions to generate the frequent pattern with the support count 10% .We have repeated the same process by increasing the transaction, after the experiment on both algorithms, we have designed a graph and summarized a result in the following table There we could see that in Apriori algorithm the time taken is directly proportion to the number of transactions where as in the Enhance apriori (after a time period the consistency of the time was maintained) take less time. In the conclusion of the analysis we can say that for 1000 records classical Apriori take 100% time while Enhance apriori take only 25% time in comparison of classical Apriori. It means that 75% time is saving. have designed a graph and summarized a result in the following table There we could see that in Apriori algorithm the time taken is

directly proportion to the number of transactions where as in the Enhance apriori (after a time period the consistency of the time was maintained) take less time. In the conclusion of the analysis we can say that for 1000 records classical Apriori take 100% time while Enhance apriori take only 25% time in comparison of classical Apriori. It means that 75% time is saving.

Table 1.1

Transactions	*Apriori*	*Enhancement in Apriori*
200	5seconds	3seconds
400	9seconds	4seconds
600	14seconds	4seconds
800	18seconds	6seconds
1000	27seconds	7seconds

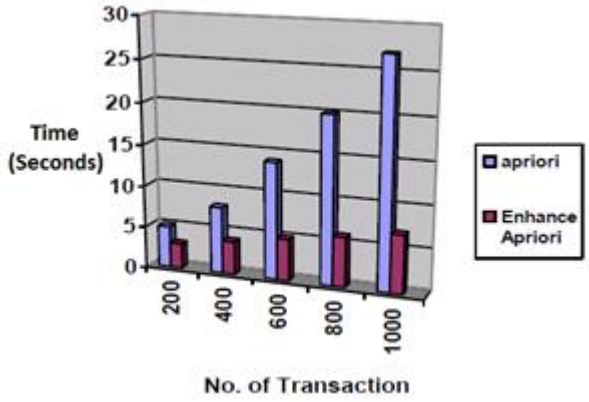

Figure 1.2: Time Comparison

5. 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 [13]. 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 filed of the data mining.

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.

[10] Arun K. Pujari.

[11] S.Prakash, R.M.S.Parvathi., An Enhanced Scaling Apriori for Association Rule Mining Efficiency. European Journal of Scientific Research, ISSN 1450-216X Vol.39 No.2 (2010), pp.257-264

[12] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. Communications of ACM, 39:58-64, 1996.

[13] G. Piatetsky-Shapiro, U. M. Fayyad, and P. Smyth. From data mining to knowledge discovery: An overview. In U.M. Fayyad, et al. (eds.), Advances in Knowledge Discovery and Data Mining, 1-35. AAAI/MIT Press, 1996.

[14] G. Piatetsky-Shapiro and W. J. Frawley. Knowledge Discovery in Databases. AAAI/MIT Press, 1991.