Volume 3, No. 11, November 2012
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

A SYSTEMATIC APPROACH AND ALGORITHM FOR FREQUENT DATA ITEMSETS

Sachin Sharma[1], Vidushi Singhal[2] and Seema Sharma[3]
[1]Deptt of Computer Applications, Manav Rachna International University,
Faridabad, Haryana, India
sachinks76@gmail.com
[2]Asst Prof, Manav Rachna International University, Faridabad
[3]Asst Prof, Manav Rachna International University, Faridabad

Abstract : The research in data mining is developing fast and efficient algorithm to derive knowledge from huge databases. There are several data mining algorithms available to solve diverse data mining problems. They are mainly classified as Associations, Classifications, Sequential Patterns and Clustering. Apriori is one of the most important algorithms used in Rule Association Mining. In this paper, we discuss the limitations of the Existing Apriori algorithm and then propose an enhancement for improving its efficiency. The drawbacks of the existing system may produce a larger number of candidate item sets and scan the database many times. The proposed algorithm is based on the reverse scan of a given database. If certain conditions are satisfied, the proposed algorithm can greatly reduce the scanning times required for the discovery of candidate itemsets. Therefore, much time and space has been saved while searching frequent itemsets.

Keywords: Frequent, Association, Support, Itemset

INTRODUCTION

The amount of data kept in computer files and databases is growing at a phenomenal rate. At the same time, the users of these data are expecting more sophisticated information from them. Knowledge Discovery in database[7] (KDD) often called Data Mining aims at the discovery of useful information from large collection of data. It is the effort to understand, analyze and eventually make use of huge volume of data available. Through the extraction of knowledge in databases, large databases will serve as a rich, reliable source for knowledge generation and verification and the discovered knowledge can be applied to information management, query processing, decision-making, process control and many other applications. Alternatively, it has been called exploratory data analysis, data driven discovery and deductive learning.

Link Analysis[3], alternatively referred to as Affinity Analysis or Association, refers to the data mining task of uncovering relationships among data. The best example of this type of application is to determine Association Rule.

Frequent Pattern Mining[4] is the most important step in mining association rules i.e. patterns showing that some items in a dataset frequently occur along with other items. It studies the frequency of items occurring together in transactional databases, and based on a threshold called Support, identifies the frequent item sets. Another threshold, confidence, which is the conditional probability than an item appears in a transaction when another item appears, is used to pinpoint association rules.

For example, it could be useful for the Video Store Manager[6] to know what movies are often rented together or if there is a relationship between renting a certain type of movies or buying popcorn or pop. The discovered Association rule are of the form : P=>Q[s,c] where P and Q are conjunctions of attribute value-pairs and s is the probability that P and Q appear together in a transaction and c is the conditional probability that Q appears in a transaction when P is present.

Association Rules can be used to assist retail management in effective advertising, marketing and inventory control.

EXISTING SYSTEM

At present, the most important association rule mining algorithm is Apriori put forward by R.Agrawal. It is an algorithm for mining the frequent itemsets.

The Apriori algorithm[1] is described as a fast algorithm for mining Association rules. The algorithm has the common structure of a two-step process. First all large itemsets that have support greater than minimum support are created incrementally. L1, the large itemsets of size 1 is created in first half over the data, by simply counting the appearance of each item in the data, Subsequent L's are created using their candidates. The candidates are potential large itemset of current size. The candidates are generated from the previous by using a fast and clever cross–product function. Then a pass over the data determines the candidate's support. All candidates with support> minsup are placed in the next L.

This process stops when Ln is empty. These large itemsets are then used to produce rules. Each large itemset is divided into all Head =>Body combinations and each combination is tested for sufficient confidence. Here confidence is sup(itemset)/ sup(head). If the confidence tested is greater than the user defined minsup then the rule Head => Body is valid and is kept in a list.

The disadvantages of existing algorithm are as follow:

Disadvantages:

a. It may produce a larger number of candidate item sets in this process.
b. It may scan the number of data scans n where n is the size of large nonempty itemset. Also the number of discovered rules is huge while most of them are non-interesting.

If all of the frequent itemsets are found, it's relatively easy to generate the association rule so the main problem is how to find all of the frequent itemsets as soon as possible.

If the maximum itemset is not frequent, it will waste time and memory space greatly.

PROPOSED SYSTEM

Based on the disadvantages of Existing Apriori Algorithm, an improved method is proposed: find the maximum frequent item sets firstly, then, get all the nonempty subsets of the frequent item set. The key of this algorithm is to find the maximum frequent item set fast.
The detailed steps are as follows:

Step 1: Construct a Structure, analyze the item sets after scanning the Database one time and put them into different addresses in the structure.
Step 2: To find all the frequent item sets, scan from the maximum item sets in the structure. Let it be k-itemset. If there are more than one k-item sets, we judge the next k-item set until getting all the maximum frequent item sets.

Step 3(a): If the k-itemset is frequent, then all its subsets are frequent (according to the property of Apriori). Include the set and all the subsets in the frequent-itemset table if not included. Go to step 6.

Step 3(b): If the k-itemset is not frequent then generate its (k-1)-itemset subsets.

Step 4: Calculate support count for all (k-1)-itemset subsets (that are not included in frequent item set table) from the structure and not by scanning the database.

Step 5: Compare each of the support count by min_sup. If the item set is frequent then goto step 3(a). If the item set is infrequent then goto step 3(b).

Step 6: Now, goto address-1 of structure and if address <0 then stop.

Step 7: Scan all the item sets if they are not included in frequent item set table then calculate the support count for them.

Step 8: Compare each support count with the min_sup. If the item set is frequent then goto step 3(a). If the item set is infrequent then got step 3(b).

Example: Take the following database of a departmental store having nine transactions. (min_sup= 2)

TID	**List of Items**
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Table 1.1: We will solve this example by using Proposed Apriori algorithm.
Step 1 -> Construct the structure. After scanning the database, we analyze three item sets and put them into different addresses 0,1 and 2.

Address	**0**		**1**		**2**	
	2-itemset	***Support***	***3-itemset***	***Support***	***4-itemset***	***Support***
Content	I2, I4	1	I1, I2, I5	1	I1, I2, I3, I5	1
	I2, I3	2	I1, I2, I4	1		
	I1, I3	2	I1, I2, I3	1		

Step 2 -> Here, Maximum Item set is 4-Itemset i.e. {I1, I2, I3, I5} and support is one, which is less than min_sup. Therefore, this item set is not frequent.
Step 3(b) -> As this 4-itemset is infrequent, therefore 3-itemset subsets are {I1, I2, I3}, {I1, I3, I5}, {I1, I2, I5} and {I2, I3, I5}.
Step 4 -> We calculate the support count of the 3-itemset subsets

In address 1 In address 2

{I1, I2, I3} = 1+1 =2
{I1, I2, I5} = 1+1 =2
{I1, I3, I5} = 1
{I2, I3, I5} = 1

Step 5 -> Here, 3-itemsets {I1, I2, I3} and {I1, I2, I5} are frequent and {I1, I3, I5} and {I2, I3, I5} are infrequent.

Step 3(a)->As {I1, I2, I3} and {I1, I2, I5} are frequent, so all its subsets are also frequent i.e. {I1, I2}, {I1, I3}, {I2, I3}, {I2, I5}, {I1, I5}, {I1}, {I2}, {I3}, {I5} are also frequent.
Step 6 -> Go to address 1 of the structure.

Step 3(b)-> As {I1, I3, I5} and {I2, I3, I5} are infrequent, so generate 2-itemset subsets are {I1, I3}, {I1, I5}, {I3, I5}, {I2, I3}, {I2, I5}.

Step 4-> Now calculate the support of 2-itemset {I3, I5}
{I3, I5} = 0 + 0 + 1 = 1

	All other subsets are already included in frequent item set.
Step 7-> As {I1, I2, I4} is not in frequent item set. {I1, I2, I4} = 1 + 0 = 1 Step 8-> As support of {I1, I2, I4} is less than min_sup so it is not frequent. Step 3(b)-> Now 2-itemset subsets of {I1, I2, I4} are {I1, I2}, {I1, I4}, {I2, I4}.	Step 5 -> As support of {i3, I5} is less than min_sup so it is not frequent. Step 3(b)-> Now the subsets of {I3, I5} are {I3}, {I5}. Step 4->As all the subsets are already included in frequent itemset, therefore stop.

Step 4-> As {I1, I2} is already included in frequent item set, therefore calculate support {I1, I4}, {I2, I4}.
{I1, I4} = 0 + 1 + 0 = 1
{I2, I4} = 1 + 1 +0 = 2
Step 5-> Now, {I1, I4} is infrequent because its support is less than min_sup and {I2, I4} is frequent because its support in greater than min_sup.

Step 3(a)->Subsets of {I2, I4} are {I2}, {I4}. As {I2} is already in frequent item set so include {I4} in frequent item set.	Step 3(b)-> Subsets of {I1, I4} are {I1}, {I4}.
Step 6-> Goto address 0	Step 4-> All the item sets are already included in frequent item set, therefore stop.
Step 7->All the item sets are already included in frequent item set, therefore stop.	

After performing all the steps, the list of frequent item sets is as follow:

1-Item set	2-Item set	3-Item set
{I1}	{I1, I2}	{I1, I2, I3}
{I2}	{I1, I3}	{I1, I2, I5}
{I3}	{I1, I5}	
{I4}	{I2, I3}	
{I5}	{I2, I4}	
	{I2, I5}	

ADVANTAGES OF PROPOSED SYSTEM

The proposed algorithm has the following advantages:

a. The database scanning is done only once to maintain the structure. Support for item sets is calculated using this structure and not by scanning the database.
b. Calculation of support of the item sets is very easy in this algorithm as compared to the Apriori Algorithm.
c. It takes less time as compared to previous Apriori algorithm.

CONCLUSIONS

This Algorithm is an improvement of Apriori, which comprehensive utilized technologies of several improved Apriori algorithm. This algorithm scans the database once. The number of scans of the database is much less than the number of scans used in existing Apriori algorithm and therefore much time and space has been saved while searching frequent item sets because it does not produce a large number of candidate item sets i.e. it avoids producing candidate item sets. So it is a low-cost and efficient method as compared to existing algorithm.

REREERENCES

Books :-

[1]. Data Mining (Concepts & Techniques), Jaiwei Han & Micheline Kamber

[2]. Modern Data Warehousing, Mining & Visualization (Core Concepts), George M. Marakas.

[3]. Data Mining (Introductory and Advanced Topics), Margaret H. Dunhan.

[4]. Data Mining, Vikram Pudi

Web Sites :-

[1]. www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm

[2]. http://www.statsoft.com/textbook/stassrul.html

[3]. http://en.wikipedia.org/wiki/Data_mining

Research Papers:-

[1]. Aggarwal R, Imielimski T, and Swami A. Mining association rules between sets of items in large databases. In proc. 1993 ACM-SIGMOD int. Conf. Management of Data, Washington DC.207-216, May 1993.

[2]. Aggarwal R and Srikant R. Fast algorithms for mining association rules. In Proc. 1994 Int. Conf. Very Large Data Bases, Santiago, Chile. 487-499, Step.1994.

[3]. XIANG-WE1 LIU, PI-LIAN HE. THE RESEARCH OF IMPROVED ASSOCIATION RULES MINING APRIORI ALGORITHM. In Proceedings of the Third International Conference on Machine Learning and Cybemetics, Shanghai, 26-29 August 2004.

