DISCOVERY OF MAXIMAL FREQUENT ITEMSET USING PRIME ALGORITHM

[1]R.SMEETA MARY, [2]DR.K.PERUMAL

[1]Assistant Professor, Department of Computer Applications, Fatima College, Madurai, India

[2]Professor, Department of Computer Applications, Madurai Kamaraj University, Madurai, India

E-mail: [1]smeetamaryr@gmail.com, [2]perumalmala@gmail.com

ABSTRACT

Data mining is the technique of discovering the new patterns in large data sets with reference to various methods at the intersection of statistics, machine learning and database systems. Computer science and statistics are the interdisciplinary subfield of data mining. The overall goal of data mining is to extort information from a data set and renew or reframe the information into a structure. Association rule is a research area in the field of data searching for frequent and using the criteria support to find frequently the items appear in the data and confidence to identify the most important relationships. In focus of this paper is to find maximal frequent itemset using a new algorithm called maximal Frequent Itemset using Prime algorithm. Most of the association rule algorithms are used to find the minimal frequent item set, and then with the help minimal frequent item set derive the maximal frequent item set. But it consumes lot of time. So to overcome this problem a new approach Maximal Frequent Itemset using Prime algorithm is proposed to find the maximal frequent item set directly. The proposed method is efficient in finding the maximal frequent item set

Keywords: *Data Mining (DM), Association Rules (AR), Frequent Itemset (FIS), Maximal Frequent Itemset using Prime algorithm (MFIPA)*

1. INTRODUCTION

Data mining is a logical process which is used to search large amount of data to find useful data. Finding out the previously unknown data is the goal of this technique. There are various steps involved such as Exploration, Pattern identification and Deployment. In exploration the data is cleaned and transformed into new form. The important variables and nature of the data based on the problem are found out. Once the data is explored using pattern identification, then it is refined and defined for the certain variables. It makes the best prediction by identifying and choosing the patterns. The last step deployment is finding out the desired outcome by deploying the patterns.

In this process the first step is Data cleaning in which noise and inconsistent data is removed and multiple data sources are combined together. In the data selection the data that are relevant for the analysis task are retrieved from the database. Data mining algorithms are used to find the extract data patterns. Some interesting measures are used for the identification of the data patterns. Using many knowledge representation techniques, knowledge is shared to the user.

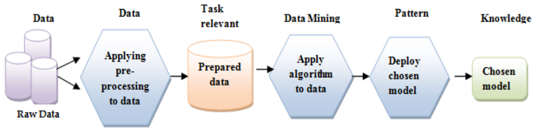

Figure 1: Process of Data Mining

At a basic level, association rule mining involves various learning models to analyze data for patterns, or co-occurrence, in a database. Association rule mining is divided into two parts that are antecedent and a consequent. An antecedent is an item which is found within the data. A consequent is an item which is found in combination with the antecedent. By using both antecedent and a consequent the association rules are created for searching the data. Hence Support and Confidence plays a vital role in identifying the relationships. Support indicates how frequently the item appears in the data and confidence indicates how many times the condition is true. In 2000, J. C. Fernando Berzal promoted an efficient method for association rule mining in relational databases [2]. This algorithm is used to find the frequent sets whose support will always be greater than minimum support. The strong association rule generates frequent item set that satisfy minimum confidence and support. But it has various disadvantages that this algorithm needs more complexity and scans the transaction database many times. In current research these issues are the disadvantages. In this paper, faster and more efficient algorithm PRIMA is promoted.

2. THEORETICAL CONCEPTS

Association Rule: Association rule mining is a method to discover correlations, frequent patterns, associations, or causal structures from data sets found in a variety of kinds of databases such as transactional databases, relational databases and other forms of data repositories.

Given a set of transactions, association rule mining intend to find the rules which enable us to expect the occurrence of a specific item based on the occurrences of the other items in the transaction.

Frequent itemsets: A set of items that materialize in many transactions is understood to be "frequent". Frequent itemset is that itemset whose support must be greater than or equal to a minimum support threshold. Frequent mining is creation of association rules on or after a transaction

3. RELATED WORK

There are number of attempt to find maximal frequent itemsets. In [3], they focused on parallel algorithms which are used to discover either maximal itemsets or frequent or closed frequent in order to solve the performance worsening, load balancing and scalability dispute of sequential algorithm. In [4], AIS algorithm was first proposed by Agarwal and Swami for mining association rule. In AIS algorithm the databases are scanned many times to get the frequent itemsets. During the first pass over the database the support count of each individual item was calculated. Item whose count is less than its minimum value are eliminated from the list of items with respect to the threshold of support count. In the second pass over the database, the support count of those candidate 2 itemsets are accumulated and checked against the support threshold. AIS algorithm has competence problems so some modifications have been introduced to give estimation for candidate itemsets that have no hope to be large. In [5], Apriori algorithm, the name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemsets properties. Apriori uses an iterative approach known as level wise search, where k itemsets are used to explore k+1 itemsets. First the frequent 1 itemsets is found, this is denoted by L1, frequent2 itemset L2 and so on. Two step processes is used to find LK, one is the join step and other is prune step to find the frequent itemset. In [6], An MFIMiner algorithm uses breadth-first search with resourceful pruning method that expertly mines both long and short maximal frequent itemsets. In [7], this search is used only for maintaining and updating a new data structure of the maximum frequent candidate set. It is used to prune candidates in the bottom up search. The very important distinctive of this algorithm is that it does not require any examination of every frequent itemset explicitly. So it performs well even when some maximal frequent itemsets are long.In [8], GenMax algorithm, which uses backtrack search method for mining maximal frequent itemset. It uses various optimizations methods to prune the search space. Progressive focusing technique is used to perform maximality checking, and different set of propagation to perform fast frequency computation. It is found that GenMax to be a highly efficient method to mine the exact set of maximal patterns. In [9] a one-pass algorithm called Data Stream Mining for Maximal Frequent Itemsets which gets the set of all maximal frequent itemsets over data streams. A data structure called summary frequent itemset forest is developed for incrementing and maintaining the information of obtaining maximal frequent itemsets in the stream.

3.1 Maximal Frequent Itemset using Prime Algorithm

Most of the algorithms used for mining maximal frequent itemsets perform fairly well when the length of the maximal frequent itemset is small. However, performance degrades when the length of the maximal frequent itemset is large. The Maximal Frequent Itemset Using Prime Algorithm proposes a new approach for mining maximal frequent itemsets. It reduces the complexity by counting the items in the horizontal method and arranging the elements in the descending order for generating maximal itemsets. The search starts by assigning the prime value to all the transactions. The values are being assigned from 1-itemset and proceeds upto n-itemsets. According to the preference the values are subtracted from the total count from n itemsets and proceeds upto 1-itemset. This search identifies the maximal frequent itemsets by examining its candidates that leads to 0. The search using this algorithm moves one-level up during a single pass whereas top-down search or bottom up approach moves many levels down during a single pass.

4. PROBLEM DESCRIPTION

4.1 Procedure

Step 1: Collect the input in individual list and put them into Hashset (Original InputSet).

Step 2: Count the individual itemcount in horizontal manner, and put it in HashSet (CountSet).

Step 3: Sort the Individual items in the HashSet by values in descending order.

Step 4: Assign consecutive prime numbers to individual items, SubstituteValSet.

Step 5: Apply prime numbers to all the individual items present in the transaction and find the total, SubstituteVal.

Step 6: while (SubstituteVal>0), Find the index of each element in an arraylist, sortedByCount and find the SubstituteValueSet of that particular element.

Step 7: while (HashSet(CountSet)>0), Subtract SubstituteValSet from SubstituteVal of all the transactions and store the value in HashSet(Qualified ItemCount) and add the element name, freqelement[].

Step 8:if any of the transactionnHashSet(Qualified ItemCount)=0 then stop the process else repeat step5 and 6.

4.2 Procedure explained with the example

Here 6 transactions and 5 items A, C, D, T and W to find maximal frequent itemset shown in below table.

Step 1: Collect the input in individual list and put them into Hashset(Original InputSet)

Table 1. Original Dataset

Tid	Titems				
T1		B	C	D	E
T2	A		C	D	E
T3		B	C	D	E
T4	A		C	D	E
T5	A	B	C	D	
T6		B	C		E
…	…	…	…	…	…
T294	A	B	C	D	E

Step 2: Count the individual item count presents in all transaction in horizontal manner and put in a two-dimensional array, HashSet. Once item frequencies are determined the algorithm will prepare the ignore list which includes a set of item not be considered in the further analysis.

Table 2. Individual Item Count

Titems	Count
A	198
B	213
C	294
D	175
E	270

Step 3: Sort the Individual items in the HashSet by values in descending order.

Table 3. Arrange in descending order

Titems	Count
C	294
E	270
B	213
A	198
D	175

Step 4: Assign consecutive prime numbers to each item in the dataset except the ones in the ignore list. Prime number assigning is done with respect to items frequency, SubstituteValSet.

Table 4. Substitute prime values

Titems	Prime values
C	2
E	3
B	5
A	7
D	11

Step 5: Apply prime numbers to all the individual items present in the transaction and find the total, SubstituteVal.

Table 5. Apply prime numbers

Titems					Total Count
	5	2	11	3	21
7		2	11	3	23
	5	2	11	3	21
7		2	11	3	23
7	5	2	11		25
	5	2		3	10
…	…	…	…	…	…
7	5	2	11	3	28

Table 6. Process the data

Tra	Tot	C	CE	CEB	CWBA
T1	21	19	16	11	4
T2	23	21	18	13	6
T3	21	19	16	11	4
T4	23	21	18	13	6
T5	25	23	20	15	8
T6	10	8	5	0	-
…	…	…	…	…	…
T294	28	26	23	18	

Step 6: While (SubstituteVal>0), Find the index of each element in an arraylist, sortedByCount and find the SubstituteValueSet of that particular element.

Step 7: While (HashSet(CountSet)>0), Subtract SubstituteValSet from SubstituteVal of all the transactions and store the value in HashSet(Qualified ItemCount) and add the element name, freqelement[].
Step 8: if any of the transaction HashSet(Qualified ItemCount) =0 then stop the process else repeat step 5 and 6. Here CEB,CEBA are considered as the maximal frequent item set.
Step 9: Check the support with the frequentelements. If suppose the support is 196 then CEB is displayed.

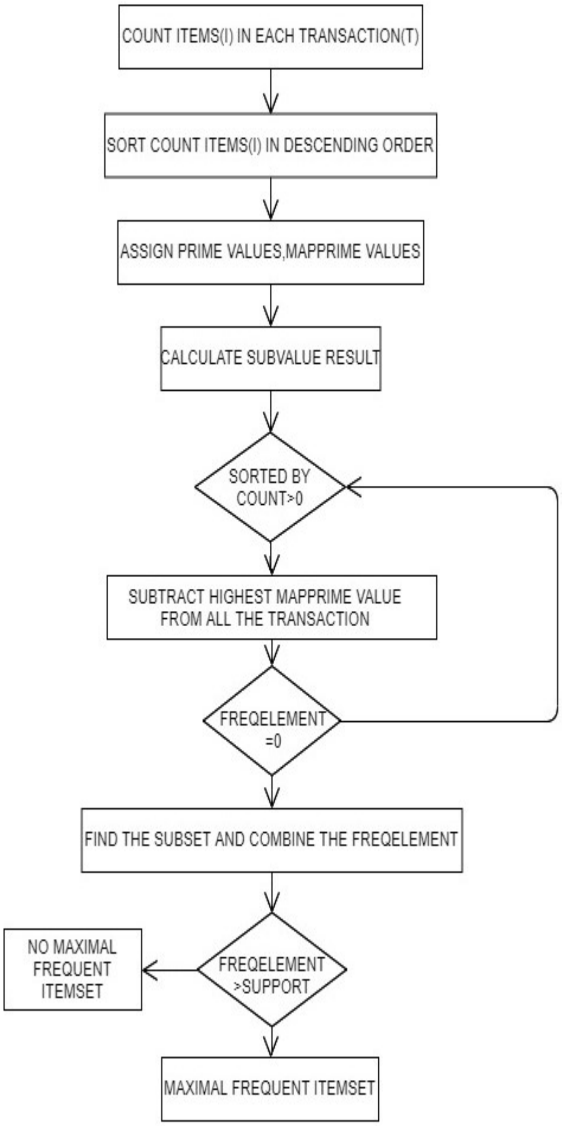
COUNT ITEMS(I) IN EACH TRANSACTION(T)
SORT COUNT ITEMS(I) IN DESCENDING ORDER
ASSIGN PRIME VALUES,MAPPRIME VALUES
CALCULATE SUBVALUE RESULT
SORTED BY COUNT>0
SUBTRACT HIGHEST MAPPRIME VALUE FROM ALL THE TRANSACTION
FREQELEMENT =0
FIND THE SUBSET AND COMBINE THE FREQELEMENT
NO MAXIMAL FREQUENT ITEMSET
FREQELEMENT >SUPPORT
MAXIMAL FREQUENT ITEMSET

Figure 2: Procedure to find maximal frequent itemsets

4.3 MFIPA

Algorithm –MFIPA
Input: dataset D, support S
Output: Maximal Frequent Itemsets MFI
MFIPA (Dataset D)
BEGIN
1. Read and store the entire preprocessed dataset.
2. Traverse horizontally through all the elements in the list and generate the sum of