Data Partitioning and Bit Vector Approach for Weighted Frequent Item Set Mining

M. A. Jabbar, B. L. Deekshatulu, and Priti Chandra

***Abstract*—Association rule mining is an important task in data mining. Association rules are frequently used by retail stores to assist in marketing, advertising, floor placement and inventory control. Most of the association rule mining algorithms will not consider the weight of an item. Weighted association is very important in KDD.In analyzing market basket analysis, people often use apriori algorithm, but apriori generates large number of frequent item sets. One alternate approach to apriori is partitioning technique. This paper presents a method to find weighted frequent item sets using partitioning and bit vector .By example, it is proved that partitioning technique can improve the efficiency by reducing the number of candidates.**

***Index Terms*—Weighted association rules, KDD, partitioning, bit vector, weighted support.**

I. Introduction

Frequent item set mining leads to the discovery of associations and correlations among item sets in large transactional data base. The information contained in data bases available to marketers has increased dramatically in the last few years. To deal with this volume of information, a new approach known as data mining has developed. Berry and linoff [1] define data mining as "The evaluation and analysis, by automatic or semi automatic means of large quantities of data in order to discover meaningful patterns and rules."

Data mining is one component of the broader process known as knowledge discovery in databases (KDD). KDD is the process of finding useful information and patterns in data.

KDD process consists of series of transformation steps 1) data cleaning2) data integration3) data selection4) data transformation5) data mining 6) pattern evaluation7) knowledge presentation

Association rule mining is very important field of research in data mining. The problem of mining association rules is put forward by R.Agrawal in 1993[2] based on support confidence frame work.

The Problem of association rule mining can be Decomposed into two sub problems. 1) Discovering frequent item sets: frequent item sets have support higher than minimal support 2) Generating rules with high confidence rules from frequent item sets

Overall performance of mining association rules are determined by first step only.

Manuscript received August 20, 2012; revised October 10, 2012.

M. Jabbar is with the JNTU Hyderabad, India (e-mail: Jabbar.meerja@gmail.com).

B. L. Deekshatulu is with the Visiting Professor HCU, Hyderabad, India.

P. Chandra is with the Scientist Advanced systems Laboratory India.

An item is said to be frequent if it satisfies minmium support threshold. The support of an item is the percentage of transactions in which that item occurs .There are various types of itemsets1) closed 2) maximal 3) emerging 4) hyper clique patterns. [3]

Apriori algorithm is the most well known association rule mining algorithm and is used in most commercial products.[4] It uses the following property, which we call the large item set property.

"Any subset of a large item set must be large"

The large item sets are also said to be downward closed because if an item set satisfies the minimum support requiremets,so do all of it's subsets. Apriori algorithm suffers from the following bottlenecks

1) Repeated number of data base scans
2) Huge candidate sets

There are various methods to improve Apriori algorithm efficiency [5]. 1) Hash based item set counting 2) Transaction reduction 3) Partitioning4) Sampling5) Dynamic item set counting.

Weight of each item represents the importance of the item in the transaction data base. Weighted association rules are derived from this weighted items based on some methods. [6] Classical association rule mining algorithms assumes that all items have same significance without considering weights

This paper finds the weighted frequent item sets using Partitioning method. The rest of this paper is organized as follows. Section 2 provides formalization of frequent item set mining algorithm, Section 3 discuss about data Partitioning approach for frequent item set mining. Method for mining weighted frequent item sets are discussed in Section 4 and Section 5 shows example. Finally we present our concluding remarks in section 6

II. Basic Concepts

Definition 1(Association rule): Given a set of items $I=\{I_1, I_2, I_{3,\ldots\ldots}I_N\}$and data base of transactions D={T1,T2…Tn}where ti=$\{I_1, I_2, I_{3,\ldots\ldots}I_N\}$,an association rule is an implication of the form $X=>Y$ where $X,Y \subseteq I$ are sets of items called item sets and $X \cap Y=\emptyset$

Definition 2(Support): the support for an association rule $X=>Y$ is the percentage of transactions in the data base that contain XUY.

$$\text{Support}(X=>Y) = P\ (\text{XUY})$$

Definition 3(Confidence): confidence or strength is the ratio of number of transactions that contain XUY to the number of transactions that contain X.

Confidence= Support (XUY)/support(X)

Definition4(Frequent item set): item set whose

occurrences is above support threshold.

Definition 5(weight of an item): The weight of each item is assigned to reflect the importance of each item in the transaction data base.

Definition6(Weighted support Degree): weighted support of rule $X=>Y$ is

$$\left(\frac{\sum_{Ti \in (x \cup y)} wi}{k} \right) \left(s\left(x \to y \right) \right)$$

III. Data Partioning Approach for Frequent Item Set

A Partitioning Technique can be used that requires just two sc1ans to mine frequent item sets. It consists of two phases. In 1st phase the algorithm subdivides transactions of Data Base D into n non overlapping partitions. If the minimum support threshold for transactions in D is min support , then the minimum support count for the partitions is min support No of transactions in that partition. For each partition all frequent item sets within the partitions are found. These are referred to as local frequent item sets.

The local frequent item sets may or may not be frequent with respect to D must occur as a frequent item set in at least one of the partitions. Therefore all local frequent item sets are candidate item sets with respect to D.

The collection of frequent item sets from all the partitions forms the global candidate item sets with respect to D. In phase 2, a second scan of D is conducted in which the actual support of each candidate is assessed in order to determine the global frequent item sets.

The main goal of this division of process into local and global computation is to process one fragment in the main memory at a time, to avoid multiple scans over D from secondary storage.

Partitioning May improve the performance of finding large item sets in several ways. [7]

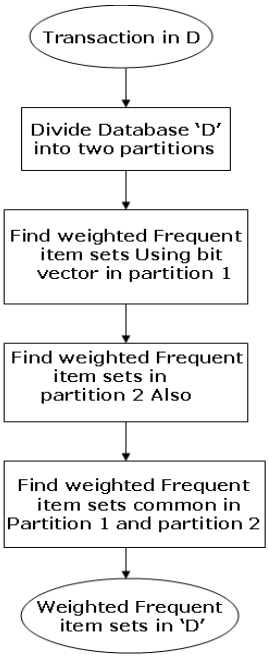

Fig. 1. Partition process

1) Parallel and distributed algorithms can be easily created, where each partition can be handled by separate machine.
2) Incremental generation of association rules may be easies to perform
3) We can design algorithms more efficiently if we adapt partitioning Technique

IV. Algorithm for Weighted Frequent Itemsets

Supermarkets/marts uses multi-dimensional data and the goods are categorized into several classes based on the type of goods. Here we are taking 5 types of goods. A stands for fruits B for Beverage, C for vegetables and D indicates food class.

Set weights to each category of goods based on importance of goods.

The transaction database for a mart shown in table I

TABLE I: Transactional Data Base

TID	Goods	TID	Goods
1	*A*1,*B*1,*B*2,*D*1	6	*A*1,*A*2,*D*1
2	*A*2,*B*1,*C*1	7	*A*2,*B*1,*B*2
3	*A*1,*A*2,*B*2,*D*1	8	*A*1,*A*2,*B*2
4	*A*1,*C*1,*D*1	9	*B*1,*B*2
5	*A*1,*A*2,*B*1,*B*2,*D*1	10	*A*1,*A*2,*C*1,*D*1

TABLE II: Weights Associated to Each Goods

Category of goods	Fruits (*A*)	Beverage (*B*)	Vegetables (*C*)	Food items (*D*)
Weight	1	0.9	0.8	0.7

Algorithm

Input: transactional data base *D*, weight of item *Wi*, support S, Weighted Support *Ws*.

Output: weighted frequent item sets

Method:

Step 1: Read the transactional data base *D*, and map into Boolean table.

Step 2:

Partition the transactional database using skipping fragments. Apply the method from step 3 to 9 for each cluster

Step 3:

Find bit vector of each item in a cluster.

Step 4:

If support ($S_{i)}$ <minimum support(*S*) then Prune item. Items which satisfy minimum support are frequent item sets.

Step 5: perform AND operation among the bit vectors of each 1 frequent item sets to generate candidate item sets.i.e to generate 2 item sets.

Step 6: find weighted support of each item sets(*X*).

Weighted support (*X*) = (W_I+W_J+------W_n)/Pattern length*Support

Step 7: if weighted support (*X*) <=minimum weighted support prune corresponding item set.

Step 8: obtain the 3- weighted frequent item sets in similar fashion.

Step 9: continue the same operation till no frequent items exist.

Step 10: combine local weighted frequent item sets in each cluster and weighted frequent item sets which are common in each cluster are global frequent item sets.

V. Example

In order to explain the algorithm we have taken the transactional data base D from table I. And let minimum support is 3. Map the above transactional data base into Boolean Matrix

TABLE III: Boolean Matrix

*A*1	*A*2	*B*1	*B*2	*C*1	*C*2	*D*1
1	0	1	1	0	0	1
0	1	1	0	1	0	0
1	1	0	1	0	0	1
1	0	0	0	1	0	1
1	1	1	1	0	0	1
1	1	0	0	0	0	1
0	1	1	1	0	0	0
1	1	0	1	0	0	0
0	0	1	1	0	0	0
1	1	0	0	1	0	1

Partition the transactional data base into 2 clusters using skipping fragments.

TABLE IV: Cluster 1

*A*1	*A*2	*B*1	*B*2	*C*1	*C*2	*D*1
1	0	1	1	0	0	1
1	1	0	1	0	0	1
1	1	1	1	0	0	1
0	1	1	1	0	0	0
0	0	1	1	0	0	0

Apply the method on cluster 1

Prune items c1 and c2 as they are not satisfying

Minimum support. Find bit vectors of each item

$BV_{A1=}$11100 $BV_{A2=}$01110 $BV_{B1=}$10111 $BV_{B2=}$11111 $BV_{D1=}$11100

*A*1, *A*2, *B*1, *B*2, *D*1 are frequent 1 item sets.Peform bit wise AND operation among 1 item sets to get 2 item sets.

$BV_{A1=}$11100 ^ $BV_{A2=}$01110 =01100 weighted support

W.S=(1+1)/2 × 2=20% so weighted support of *A*1,*A*2=20%

$BV_{A1=}$11100 ^ $BV_{B1=}$10111 =10100 weighted support of *A*1,*B*1=(1+0.9)/2 × 2=19% so weighted support of *A*1,*B*1=19%

$BV_{A1=}$11100 ^ $BV_{B2=}$11111=11100 weighted support of *A*1, B2== (1+0.9)/2×3=28% so weighted support of *A*1, *B*2=28%

$BV_{B1=}$10111 ^ $BV_{B2=}$11111 =10111 weighted support of *B*1, *B*2=36%

$BV_{A1=}$11100 ^ $BV_{D1=}$11100=10100 weighted support of *A*1, *D*1=25%

$BV_{B1=}$10111 ^ $BV_{D1=}$11100=10100 weighted support of *B*1, *D*1=16%

$BV_{A2=}$01110 ^ $BV_{D1=}$11100=01100 weighted support of *A*2, *D*1=17%

Item set *B*1, *D*1 having less minimum support percentage hence *B*1, *D*1 will be pruned.

Now find the weighted 3 item sets

First find AND operation between item sets *A*1, *A*2^ *A*1, D1 =01100 i.e. *A*1, *A*2, *D*1 and similarly *A*1, *A*2^*A*2, *D*1= 01100 weighted support of *A*1, *A*2, *D*1=(1+1+0.7)/3 × 4=36%

Similarly *A*1, *A*2^ *A*1, *B*2=01100 weighted support of *A*1, *A*2, *B*2= (1+1+0.9)/3×2=19%

Hence in the cluster 1 local weighted frequent item sets are *A*1, *A*2, and *D*1.

TABLE V: Cluster 2

*A*1	*A*2	*B*1	*B*2	*C*1	*C*2	*D*1
0	1	1	0	1	0	0
1	0	0	0	1	0	1
1	1	0	0	0	0	1
1	1	0	1	0	0	0
1	1	0	0	1	0	1

Apply the algorithm on cluster 2 also.

Prune item c2 as it is not satisfying the minimum Support. Remaining items are *A*1, *A*2, *B*1, *B*2, *C*1, and *D*1

$BV_{A1=}$01111 $BV_{A2=}$10111 $BV_{B1=}$10000 $BV_{B2=}$00010 $BV_{C1=}$11001 $BV_{D1=}$01101

Prune *B*1 and *B*2 as they are not satisfying minimum support. Hence frequent 1 item sets are *A*1, *A*2, *C*1, and *D*1

Perform the AND operation among the 1 frequent item sets $BV_{A1=}$01111 ^ $BV_{A2=}$10111 =00111 weighted support of *A*1, *A*2 is (1+1)/2×3=30%

$BV_{A1=}$01111 ^ $BV_{D1=}$01101=00111 weighted support of *A*1, *D*1 is (1+0.7)/2×3=25%

$BV_{A2=}$10111 ^ $BV_{D1=}$01101=00011 weighted support of *A*2,*D*1 is (1+0.7)/2×2=17%.From the 2 frequent weighted item sets find 3 weighted frequent item sets

$BV_{A1=}$01111 ^ $BVA2_{=}$10111 ^ $BVD1_{=}$01101 weighted support of *A*1, *A*2, *D*1= (1+1+0.7) =18%

In cluster 2 weighted frequent items set is *A*1, *A*2, and *D*1.This item set is local weighted frequent item set in cluster 2.Common weighted frequent item set in cluster 1 and cluster 2 are *A*1, *A*2, and *D*1.so *A*1, *A*2, and *D*1 are called global weighted frequent item sets.

From the transactional data base *A*1, *A*2, and *D*1 are 3 weighted frequent item sets.

VI. Conclusion

Weighted association is important in KDD. Data partitioning approach reduces number of candidates. In this paper we suggested a conceptual model that allows development of an efficient algorithm to Real world data for weighted association rule mining. The proposed algorithm produces less no. of candidate item sets, takes less time and memory space when compared with Apriori algorithm.

References

[1] Berry, J. A. Michael, and G. Linoff, *Data Mining Techniques for Marketing Sales and Customer Support*, New York, 1997

[2] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," *Proceeding of the ACM SIGMOD Intl. Conf. on Management of Data*, pp. 207-216, 1993.

[3] P.–N. Tan and M. Steinbach, *Vipin Kumar Introduction to Data Mining*

[4] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc 20th Int. Conf. Very large Data Bases*, pp. 487-499, 1994.

[5] Han and Kamber, *Data Mining Concepts*, Elsevier, 2009

[6] Y. Shaoqian, "A Kind of improved algorithm for weighted apriori and applications to data mining," *5th International Conference of Computer Mining Science and Education Hefei*, China, pp. 24-27, Aug. 2010

[7] M. H. Dunham and S. Sridhar, "Data mining introductory and advanced topics," *Pearson Education*, pp. 174-175, 2008.