International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
International Conference & workshop on Advanced Computing 2013 (ICWAC 2013) – www.ijais.org

Mining of frequent Itemset using PAFI and Transaction Reduction Method

Anil Vasoya
Assistant Professor
TCET , Kandivali(e)
Mumbai

Rekha Sharma, Ph.D
Associate Profeesor, Dy. HOD, CMPN
TCET , Kandivali(e)
Mumbai

ABSTRACT

Now a day, Mining of Association rules in large database is the challenging task. An Apriori algorithm is widely used to find out the frequent item sets from large database. But it has some limitations. It produces overfull candidates while finding the frequent item sets from transactions, i.e. the algorithm needs to scan database repetitively while finding frequent item sets. It will be inefficient in large database and also it requires more I/O load while accessing the database frequently.
To solve the bottleneck of the Apriori algorithm, PAFI and Matrix based method used in proposed system.

General Terms

Data Mining, Association rule.

Keywords

PAFI, Apriori algorithm, frequent Itemset, clustering, AND operation, affair.

1. INTRODUCTION

An Association rule plays an important role in recent data mining techniques. The purchasing of one product along with another related product represents an association rule. Association rules are used to show the relationships between data items. Association rules are frequently used in marketing, advertising and inventory control. Association rules detect common usage of items. This problem is motivated by applications known as market basket analysis to find relationships between items purchased by customers [4] [5]. That is, what kinds of products tend to be purchased together?

The associations between data are complicated and most of them are hidden. Association rule mining is the mostly used method in Association Knowledge Discovery which aim is to find out the hidden information. The most famous is the Apriori algorithm which has been brought in 1993 by Agrawal, etl [1]. But it has two deadly bottlenecks [2]:

(1) It needs great I/O load when frequently scans database.

(2) It may produce overfull candidates of frequent item sets.

To solve the bottleneck of the Apriori algorithm [2], proposed system will used PAFI (Partition Algorithm for Mining Frequent Item sets) for clustering. This algorithm partitions the database transactions into clusters. Clusters are formed based on the similarity measures between the transactions. After forming the clusters we need to find out frequent item set from each cluster using matrix based method [3].

2. LITERATURE REVIEW

2.1 Find frequent itemsets using Apriori algorithm:

The most famous is the Apriori algorithm which has been brought in 1993 by Agrawal which uses association rule mining [6].

Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. Association rule generation is usually split up into two separate steps:

1. Minimum support is applied to find all frequent item-sets in a database.
2. These frequent item-sets and the minimum confidence constraints are used to form rules.

Advantage of this algorithm, it is easy to find frequent item sets if database is small but it has two deadly bottlenecks. First, It needs great I/O load when frequently scans database and Second, It may produce overfull candidates of frequent item-sets.

2.2 Find frequent itemsets using PAFI as well as Apriori algorithm:

D.Kerana Hanirex and Dr.M.A.Dorai Rangaswamy proposed efficient algorithm for mining frequent item sets using clustering techniques. They presents an efficient Partition Algorithm for Mining Frequent Item sets (PAFI) using clustering. This algorithm finds the frequent itemsets by partitioning the database transactions into clusters and after clustering it finds the frequent itemsets with the transactions in the clusters directly using improved Apriori algorithm which further reduces the number of scans in the database as well as easy to manage and available easily, hence improve the efficiency as well as new algorithm better than the Apriori in the space complexity but again it uses apriori algorithm hence efficiency not increase as much as required.

2.3 Find frequent itemsets using Improved Apriori algorithm based on matrix:

Feng WANG and Yong-hua proposed An improved Apriori algorithm based on the matrix. To solve the bottleneck of the Apriori algorithm, they introduce an improved algorithm based on the matrix [8]. It uses the matrix effectively indicate the affairs in the database and uses the “AND operation” to deal with the matrix to produce the largest frequent itemsets and others. The algorithm based on matrix don’t scan database frequently, which reduce the spending of I/O. So the new algorithm is better than the Apriori in the time complexity but it is not suitable for large database.

Its understand that PAFI algorithm is better for partitioning large database and Matrix method is better for find out frequent item set from cluster hence we use mixture of PAFI and Matrix based method so we can achieved better space complexity using PAFI algorithm and better time complexity using Matrix based method..

3. PROPOSED ARCHITECTURE

To solve the bottleneck of the Apriori algorithm [2] i.e. it needs great I/O load when frequently scans database and it produce overfull candidates of frequent item sets so it is challenge to reduce the number of scan or reduce the time also challenge to the space requirement of main memory.

3.1 Problem Definition

General idea used are that reduce passes of transaction database scans and shrink number of candidates so that it is easily fit into main memory even if database is large. Hence to reduce the number of candidate it is proposed that , We partition the whole database in to different cluster using PAFI algorithm after finding out the clusters. Then in second phase matrix method of transaction reduction [3] applied on each cluster so that we do not need to scan database again.

In propose system proposed combination two methods are used. First, partition the large database into different cluster so the better space complexity achieved and then find frequent item sets from all the cluster using matrix method so here it achieved better time complexity and thus it can overcome from both the drawback of apriori algorithm.

In proposed system combine two algorithms called PAFI and Matrix based algorithm and which follow the following steps:

i. For a set of transactions in the database D, it applies partition algorithm in order to find clusters based on the number of transactions. We are getting clusters CL1, CL2 and so on.

ii. Partition the whole databases regardless to the size of cluster but cluster size is less than main memory size.

iii. After finding out all the Clusters apply matrix based method on each cluster and find out the frequent item tests from each cluster.

iv. List out all the frequent item sets of all the clusters of whole database.

4. IMPLEMENTATION METHODO LOGY

The entire database divided into partitions of variable sizes using PAFI algorithm,

PAFI (Partition Algorithm for Frequent Itemsets) ALGORITHM

```
Begin
Number of clusters (NOC) =count of transactions (COT)/N
// where N is random natural number
FOR i= 1 to NOC DO BEGIN
FOR each cluster Ci DO BEGIN
FOR each transaction t €D DO BEGIN
Find t such that t having highest number of items
Put t in Ci
END
END
Return Clusters with  itemset.
```

Now each partition will be called a cluster. Each cluster is considered one at a time by loading the first cluster into memory and calculating frequent itemset using transaction reducation algorithm and the corresponding support counts. Then the second cluster is considered similarly and the cumulative support count is calculated for the cumulative large itemsets. This process is continued for the entire set of clusters and finally we have the whole large itemsets and the corresponding cumulative support counts. This approach reduces main memory requirement since it considers only a small cluster at a time and hence it is scalable for any large size of the database.

For finding the large itemsets it is enough to go through the transactions with in the clusters. There is no need to go through the entire database again. Hence it reduces the redundant database scan and improves the efficiency.

Process description of transaction reduction algorithm:

The detailed process of the algorithm based on the matrix as follows:

(1) Convert the affair database to a matrix and operate it.

First, convert the affair database contained i items and t affairs to a matrix which has i+1 rows and i+1 1 columns, the fist column notes items and the first row notes affairs. As shown in TABLE.

Second, according to the minsupport simplified the matrix. When the sum of an item less than the minsupport, we can know that the item will not be in any frequent itemsets, so we should delete the row contained the item, which will simplify the later operation.

The matrix through the step is marked as the initial matrix "Mat" which is the base of the later operation.

(2) Finding out the largest K-frequent itemsets uses the way of operating the matrix.

The largest frequent itemsets is a frequent itemsets which contains the most items than others. According the nature 1, it will be easier to find the frequent itemsets contained fewer items if we know the largest frequent itemsets.

There are several steps:

(a) Finding out the affair in the matrix which has the most items and the number of items is marked as K. If the number of affairs which the number of the items not less than the K less than the minsupport, which means the largest frequent itemsets cannot contain K items, so we should consider the affair which has K-1 items and use the K to mark the K-1.use this method until we find out the K that the number of affairs is not less than the minsupport.

(b) We use recursive algorithm to simplify the Mat, there are several steps: i) Deal with the column: because what we should judge is the support of the frequent itemsets contained K items, we should delete the column which the number of items less than the K. ii) Deal with the row: through the step a), the row may have be changed. So, we should delete the row which the number of affairs less than the minsupport. iii) Through the step ii, the column may have been changed. We should go to the step i. If the i and ii have not changed the Mat, we should mark the Mat as NewMat and go to the step iii. If the Mat is empty, which means there aren't K-frequent itemsets, so we should go to the step i and consider the affair which has K-1 items.

(c) First, we select an affair "t" in the NewMat and select K items in the t to form an itemsets. Second, Using the "AND operation" count the Support of the itemsets. If its support is not less than the minsupport, the itemsets is a K-frequent itemsets. Or else it is not. Then we again select the K items in

the t which is not the same as the former and judge the support. Use this method until there is no the different K items in the t. Now, we can delete the column contained the t and simplify the NewMat using the same way as the step (b). Then, we use the same way to deal with the other affairs in the NewMat and the itemsets which has been counted should not be considered again. Use this way until the NewMat is empty. If there are not K-frequent itemsets,we should go to the step (a) and consider the affair which has K-1 items.
(3)Using the Mat finds out all the other frequent itemsets from 2-frequent itemsets to K-1-frequent itemsets, and the method is similar to the Apriori algorithm. But we should simplify the Mat which made counting the support easier.
According to the nature 1, because we have know the K-frequent itemsets, which contains a mass of K-a-frequent itemsets($0<a<K-1$), we can reduce the amount of candidates of frequent itemsets, which means it will reduce the number of counting Support, connecting and pruning, which is the Apriori algorithm's problem.

The algorithm based on matrix donot scan database frequently, which reduce the spending of I/O. So the new algorithm is better than the Apriori in the time complexity as well as we partition the whole database into clusters hence space or memory complexity also better than apriori algorithm.

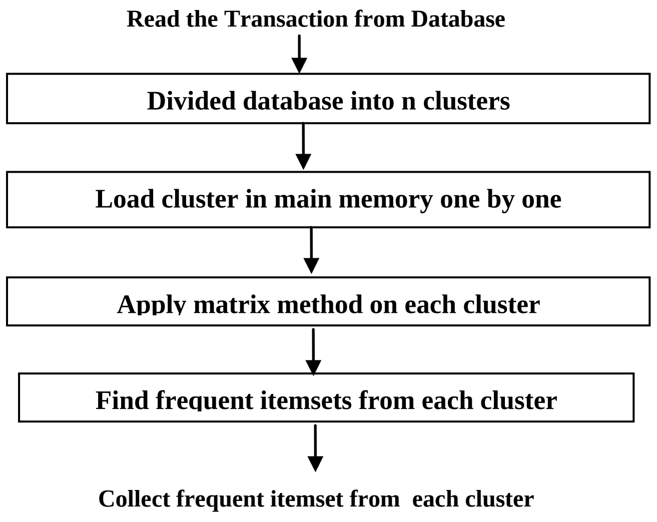

Read the Transaction from Database
Divided database into n clusters
Load cluster in main memory one by one
Apply matrix method on each cluster
Find frequent itemsets from each cluster
Collect frequent itemset from each cluster

Figure 1 : Process description of proposed system

Hence time required to finding out frequent itemsets required less time than apriori . Instead of whole database only clusters come into main memory so it can be easily fit into main memory.

5. RESULTS & DISCUSSION

To find frequent item sets from large database we proposed combination two method first, partition the large database into different cluster so the better space complexity achieved and then find frequent item sets from all the cluster using matrix method so we can achieved better time complexity and we can overcome from both the drawback of apriori algorithm.
Here we combine two algorithms called PAFI and Matrix based algorithm and which follow the following points:
Steps:

i. For a set of transactions in the database D, it applies partition algorithm in order to find clusters based on the number of transactions. We are getting clusters CL1, CL2 and so on.
ii. Partition the whole databases regardless to the size of cluster but cluster size is less than main memory size.
iii. After finding out all the Clusters apply matrix based method on each cluster and find out the frequent item tests from each cluster.
iv. List out all the frequent item sets of all the clusters of whole database.

If N is the number of transactions in the affair database, T is the longest transaction, in the Apriori algorithm, the time of the first time scan database is $O(T\times N)$, the time of connecting is $O(|Lk-1|\times|Lk-1|)$, the time of pruning is $O(|Ck|)$, the time of counting support is $O(N\times|Ck|)$. So the total time of the Apriori as follows [3] [7]:

$$O(T\times N)+\Sigma(O(|Lk-1|\times|Lk-1|)+O(|Ck|)+O((N\times|Ck|))(K>1) \quad \ldots.(1)$$

In the algorithm based on matrix, the time of finding the largest frequent itemsets, which is:

$$O(|T\text{-}MaxK|\times|T\text{-}MaxK|\times|LMsxK|).$$

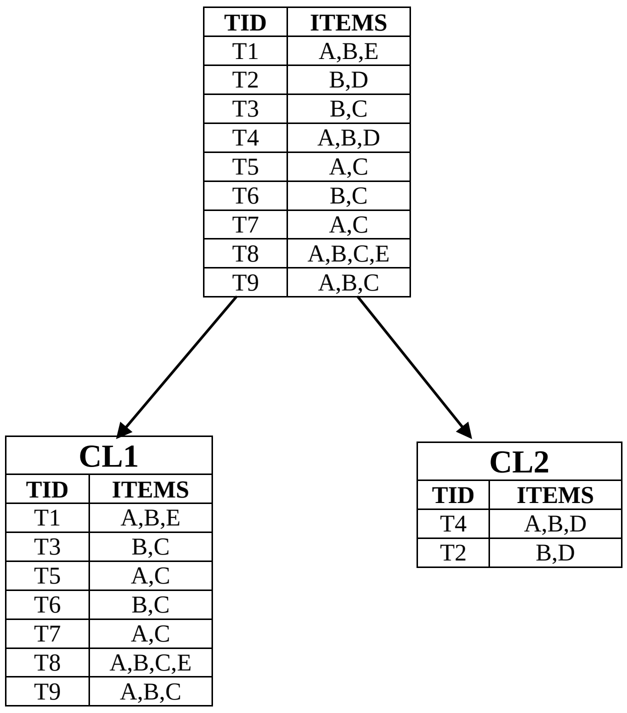
So the total time of it as follows:

$$O(T\times N)+O(|T\text{-}MaxK|\times|T\text{-}MaxK|\times|LMsxK|)+\Sigma(O(|LK-2|\times|LK-2|)(1<K<MaxK)\ldots\ldots\ldots\ldots.(2)$$

Although in the second formula there is the time of LK is far less than the Lk in the first, and CK in the second also is far less than the CK in the first. So here we did not scan whole database again hence achieved better time and space complexity.

6. EXPERIMENTAL RESULTS

This is an example based on the following transactions in the database D. First we are applying Partition algorithm(PAFI) to find clusters then applying transacton reduction algorithm to find the frequent itemsets.
Steps:
1. For a given set of transactions in the database D, it applies partition algorithm in order to find clusters based on the number of transactions. Here we are getting 2 clusters CL1 and CL2.



TID	ITEMS
T1	A,B,E
T2	B,D
T3	B,C
T4	A,B,D
T5	A,C
T6	B,C
T7	A,C
T8	A,B,C,E
T9	A,B,C

CL1

TID	ITEMS
T1	A,B,E
T3	B,C
T5	A,C
T6	B,C
T7	A,C
T8	A,B,C,E
T9	A,B,C

CL2

TID	ITEMS
T4	A,B,D
T2	B,D

Figure 2: Set of transactions in the database with partition

2. After forming cluster using PAFI algorithm, now apply the transaction reduction algorithm on each cluster i.e. CL1 and CL2 but here CL2 has less number of transactions that is less than the threshold value so we are deleting the transactions in CL2 and applying transaction reduction algorithm only on the transactions in CL1.

As shown in Figure 2, the affair cluster i.e. CL1 has 7 affairs, CL1={T1,T3,T5,T6,T7,T8 ,T9}, the itemsets is I={A,B,C,E} and the minsupport is 2.

TID	ITEMS
T1	A,B,E
T3	B,C
T5	A,C
T6	B,C
T7	A,C
T8	A,B,C,E
T9	A,B,C

Figure 3: Set of transactions in Cluster 1

(A) Find out the Mart of CL1

Transaction / Item	T1	T3	T5	T6	T7	T8	T9
A	1	0	1	0	1	1	1
B	1	1	0	1	0	1	1
C	0	1	1	1	1	1	1
E	1	0	0	0	0	1	0

Figure 4: Mart of CL1

As shown in figure 4, create the matrix according the affair cluster. If an item in an affair, the position was set 1, or else set 0.

There is no one row has "1" less than the minsupport 2, so we should not delete any row.

(B) Find out the largest frequent itemsets L_{MaxK}

We can find out the largest frequent itemsets by simplifying the above figure 4 Operations as follows:
(1) As shown in figure 4, the number of the most items in an affair is 4, but only an affair "T8" has 4 items, so the number of affairs had 4 items is less than the minsupport "2". But there are 3 affairs have 3 items or more: {T1, T8, T9}

Transaction / Item	T1	T8	T9
A	1	1	1
B	1	1	1
C	0	1	1
E	1	1	0

Figure 5: 3 affairs after reduction of T08

(2) We should simplify the matrix according 3 items. As shown in figure , the affair "T1" has an itemsets contained 3 items {A,B,E}, we do the "AND operation" to the rows "A", "B", "E".

Transaction / Item	T1	T8	T9
A	1	1	1
B	1	1	1
E	1	1	0
Result of "AND" operation	1	1	0

Figure 6: Result of "AND" operation on {A,B,E}

The result is 2 which is no less than the minsupport "2", so the itemsets {A, B, E} is one of the frequent itemsets.

Again the affair "T9" has an itemsets contained 3 items {A, B, C}, we do the "AND operation" to the rows "A", "B", "C".

Transaction / Item	T1	T8	T9
A	1	1	1
B	1	1	1
C	0	1	1
Result of "AND" operation	0	1	1

Figure 7: Result of "AND" operation on {A,B,C}

The result is 2 which is no less than the minsupport "2", so the itemsets {A, B, C} is also one of the frequent.

Hence {A, B, E} and {A, B, C} are frequent itemset findout from cluster1 i.e. CL1.

6. CONCLUSION

In this paper, The entire database divided into partitions of variable sizes, each partition will be called a cluster. Each cluster is considered one at a time by loading the first cluster into memory and calculating large itemsets and the corresponding support counts. Then the second cluster is considered similarly and the cumulative support count is calculated for the cumulative large itemsets. This process is continued for the entire set of clusters and finally we have the whole large itemsets and the corresponding cumulative support counts. This approach reduces main memory requirement since it considers only a small cluster at a time and hence it is scalable for any large size of the database.

For finding the large itemsets it is enough to go through the transactions with in the clusters. There is no need to go through the entire database again. Hence it reduces the redundant database scan and improves the efficiency.

The algorithm based on matrix donot scan database frequently, which reduce the spending of I/O. So the new algorithm is better than the Apriori in the time complexity as well as we partition the whole database into clusters hence space or memory complexity also better than apriori algorithm.

7. REFERENCE

[1] Agrawal R, Imielinski T, Swami A, “Mining association rules between sets of items in large databases”. In: Proc. of the l993ACM on Management of Data, Washington, D.C, May 1993. 207-216

[2] D.Kerana Hanirex, Dr.M.A.Dorai Rangaswamy:” Efficient algorithm for mining frequent item sets using clustering techniques.” In International Journal on Computer Science and Engineering Vol. 3 No. 3 Mar 2011. 1028-1032

[3] Feng WANG, Yong-hua LI:”Improved apriori based on matrix”,IEEE 2008, 152-155.

[4] Han Jiawei, Kamber Miceline. Fan Ming, Meng Xiaofeng translation, “Data mining concepts and technologies”. Beijing: Machinery Industry Press. 2001

[5]Margatet H. Dunham. Data Mining, Introductory and Advanced Topics: Upper Saddle River, New Jersey: Pearson Education Inc.,2003.

[6] Chen Wenwei, “Data warehouse and data mining tutorial”. Beijing: Tsinghua University Press. 2006

[7] Tong Qiang, Zhou Yuanchun, Wu Kaichao, Yan Baoping, “ A quantitative association rules mining algorithm”. Computer engineering. 2007, 33(10):34-35

[8] Zhu Yixia, Yao Liwen, Huang Shuiyuan, Huang Longjun, “ A association rules mining algorithm based on matrix and trees”. Computer science. 2006, 33(7):196-198

[9] Wael A. AlZoubi, Azuraliza Abu Bakar, Khairuddin Omar,” Scalable and Efficient Method for Mining Association Rules”, International Conference on Electrical Engineering and Informatics 2009.

[10] Wael Ahmad AlZoubi, Khairuddin Omar, Azuraliza Abu Bakar “An Efficient Mining of Transactional Data Using Graph-based Technique”,3rd Conference on Data Mining and Optimization (DMO) 2011, Selangor, Malaysia