International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
International Conference & workshop on Advanced Computing 2014 (ICWAC 2014) – www.ijais.org

Novel Approach to Improve Apriori Algorithm using Transaction Reduction and Clustering Algorithm

Anil Vasoya
Assistant Professor ,
Thakur College of Engg. &
Technology, Mumbai, India

ABSTRACT
Now a day, Association rules mining algorithms used to increased turnover of any product based company. Therefore, many algorithms were proposed to determine frequent itemsets. This paper also proposes a novel algorithm, which is resulting from merging two existing algorithms (i.e. Partition with apriori and transaction reduction algorithm) to derived frequent item sets from large database. The experiments are conducted to find out frequent item sets on proposed algorithm and existing algorithms by applying different minimum support on different size of database. It shows that designed algorithm (pafi with apriori algorithm) takes very much less time as well as it gives better performance when there is a large dataset. Whereas with increase in dataset, Apriori and Transaction reduction algorithm gives poor performance as compared to PAFI with apriori and proposed algorithm. The implemented algorithm shows the better result in terms of time complexity. It also handle large database with efficiently than existing algorithms.

General Terms
Apriori algorithm, frequent Itemset (FIS)

Keywords
PAFI, , clustering, Transaction reduction

1. INTRODUCTION
Now a days due to rapid growth of data in organizations, large scale data processing is a focal point of information technology. Mining of Association rules in large database is the challenging task. An Apriori algorithm [1] is widely used to find out the frequent item sets from database.

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 for different purposes like marketing, advertising and inventory mart. Association rules find out 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 and Matrix algorithm to find frequent item set from each cluster. 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 sets from each cluster using matrix based method [3] with less amount of time. Hence the main goal of the recommended system is to improve time complexity.

2. LITERATURE SURVEY
Mining of frequent pattern is the mining of the frequently occurring ordered events or subsequences as patterns. An example of frequent item set is pencil, eraser & sharpener because “Customers who purchase a pencil are likely to buy eraser or sharpener”. Now a day’s many algorithms available to find the frequent item set from database.

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 (threshold) 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 because of partition each cluster or partition easily swap in or swap out as well as Matrix method is better for find out frequent item set from each cluster with less span of time hence by using mixture of PAFI and Matrix based algorithm, it is easy to achieved frequent item set with better time and space complexity.

3. PROPOSED WORK

To solve the bottleneck of the Apriori algorithm [2] i.e. it needs great I/O load when frequently scans database and it produces overfull candidates of frequent item sets so it is challenging to reduce the number of scans their by reducing the time and main memory requirement.

3.1 Problem Definition

General idea used is to reduce number of 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 to, divide the whole database in to different cluster using PAFI algorithm After finding out the clusters, matrix method of transaction reduction [3] is applied on each cluster so that it do not need to scan database again.

3.2 System Architecture

Propose algorithm uses two existing algorithms. In the beginning it uses PAFI for clustering and then Matrix method on each cluster. It shows in the figure 1.

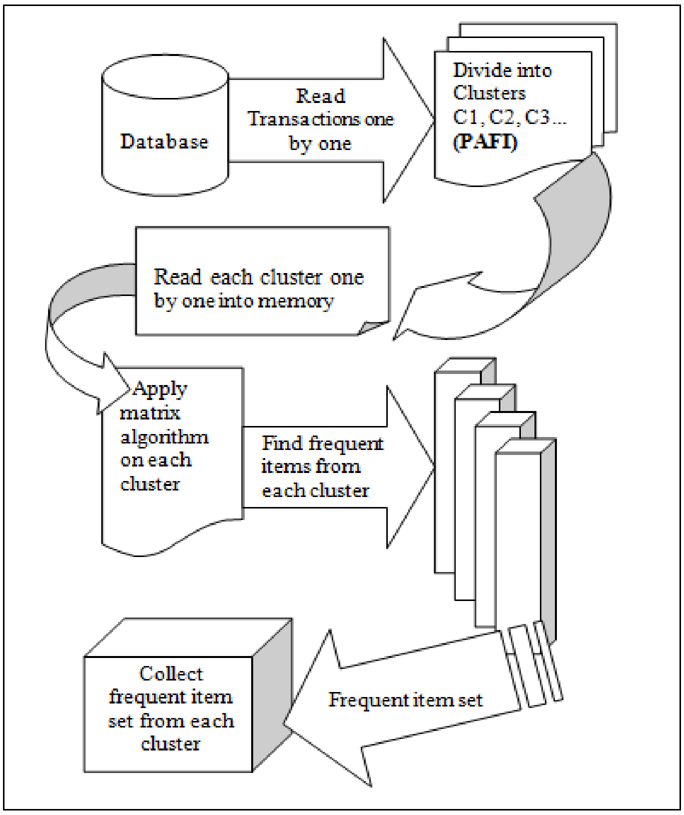


Figure 1. Outline of the proposed algorithm

In proposed algorithm, First, large database partition e into different clusters to achieved better space complexity and then frequent item sets are found from all the clusters using matrix method for achieving better time complexity and thus it can overcome from both the drawback of apriori algorithm.

In proposed algorithm combine two algorithms called PAFI and Matrix based algorithm used. Below algorithm shows the steps of proposed algorithm.

Algorithm :
Input: Database, Threshold and Number of clusters.
Output: Generate clusters, matrix and frequent item sets
Steps:-

1. Given set of transaction in the database.
2. Read Number of clusters.
3. Arrange all transaction in descending order, put it in the list.
4. As per input of number of cluster, select that many transactions in the list from the top and place it on the first position of every cluster.
5. After selection of first transaction in every clusters scan all transaction one by one and put highest similarity or minimum 3 similar items transaction in that cluster.
6. Step 5 will repeat till all transaction will be scanned.
7. Select next cluster from the list and repeat step 5 and 6.

8. Generate all clusters as per input.
9. Convert first cluster into matrix form (Mart).
10. First column notes items available in that cluster and row notes all transaction number of that cluster.
11. After forming first row and column, if item of particular transaction present than marked as '1' otherwise marked as '0' in the matrix.
12. Find out all items of that matrix (K) then put it into the list and find out Number of transactions (N) consist of that K items by applying AND operation.
13. If N > threshold than K is a frequent item set otherwise not.
14. Then consider different combination of K - 1 item as much as possible.
15. Go to step 13 till found all frequent item set of that cluster.
16. Now take next matrix into the memory and repeat step 10 to 15 till get frequent item sets from all matrixes.
17. End

In proposed algorithm, number of clusters, number of minimum similar items from transaction and minimum support threshold is decided by user. After that the entire database divided into that many clusters. After generating the cluster the clusters that have the total number of transactions less than some threshold value will be deleted.

Now it is easy to apply matrix algorithm on each cluster rather than applying matrix algorithm on entire database. Cluster will required less space hence memory complexity also increases. It is easy to find out frequent item sets from cluster than entire database

After applying matrix algorithm on each matrix, generate FIS (frequent item set from all matrix (all clusters) and arrange frequent item set of all cluster in to the array.

4. EXPERIMENTAL RESULTS

This section includes two examples which are solved using proposed algorithm, the performance analysis of different size of dataset using proposed algorithm with existing three algorithms. The purpose is to observe, the performance of various algorithm with increase in number of transactions.

Example 1: For a given set of transactions in the database D, which consist of only 9 transaction and 5 items and it divided into two clusters.

Table 1: Database

TID	ITEMS
T1	1,2,5
T2	2,4
T3	2,3
T4	1,2,4
T5	1,3
T6	2,3
T7	1,3
T8	1,2,3,5
T9	1,2,3

2. Above database divided into two clusters as show in below table 2.

Table 2: Set of transactions in the database with partition

Cluster 1	
TID	**ITEMS**
T1	1,2,5
T3	2,3
T5	1,3
T6	2,3
T7	1,3
T8	1,2,3,5
T9	1,2,3

Cluster 2	
TID	**ITEMS**
T4	1,2,4
T2	2,4

3. After forming cluster using PAFI algorithm, now apply the transaction reduction algorithm (matrix) 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 item sets is I={1,2,3,4,5} and the minsupport (Threshold) is 2.

Table 3: Set of transactions in Cluster 1

TID	ITEMS
T1	1,2,5
T3	2,3
T5	1,3
T6	2,3
T7	1,3
T8	1,2,3,5
T9	1,2,3

(A) Find out the Mart of CL1

Table 4: Mart of CL1

Transaction / Item	T1	T3	T5	T6	T7	T8	T9
1	**1**	**0**	**1**	**0**	**1**	**1**	**1**
2	**1**	**1**	**0**	**1**	**0**	**1**	**1**
3	**0**	**1**	**1**	**1**	**1**	**1**	**1**
5	**1**	**0**	**0**	**0**	**0**	**1**	**0**

As shown in table 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 threshold 2, so we should not delete any row.

(B) **Find out the largest frequent itemsets**

find out the largest frequent itemsets by simplifying the above table 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 threshold "2". But there are 3 affairs have 3 items or more: {T1, T8, T9}

Table 5: 3 affairs after reduction of T08

Transaction / Item	T1	T8	T9
1	**1**	**1**	**1**
2	**1**	**1**	**1**
3	**0**	**1**	**1**
5	**1**	**1**	**0**

(1) We should simplify the matrix according 3 items. As shown in figure , the affair "T1" has an itemsets contained 3 items {1,2,5}, do the "AND operation" to the rows "1", "2", "5".

Table 6: Result of "AND" operation on {1,2,5}

Transaction / Item	T1	T8	T9
1	1	1	1
2	1	1	1
5	1	1	0
Result of "AND" operation	**1**	**1**	**0**

The result is 2 which is no less than the threshold "2", so the itemsets {1, 2, 5} is one of the frequent itemsets. Again the affair "T9" has an itemsets contained 3 items {1, 2, 3}, we do the "AND operation" to the rows "1", "2", "3".

Table 7: Result of "AND" operation on {1,2,3}

Transaction / Item	T1	T8	T9
1	1	1	1
2	1	1	1
3	0	1	1
Result of "AND" operation	**0**	**1**	**1**

Now apply partition algorithm (PAFI) in order to find clusters based on the number of transactions. Given input as number of cluster = 9.

After applying the PAFI algorithm entire database of 50 transactions divided in to nine clusters as per the modified algorithm.

The result is 2 which is no less than the threshold "2", so the itemsets {1, 2, 3} is also one of the frequent.
Hence {1, 2, 5} and {1, 2, 3} are frequent itemset findout from cluster1 i.e. CL1.

Example 2:

Table below shows a given set of 50 transactions in the database D

Table 8: Set of transactions in the database

TID	Items
T01	1,3,5,7,9
T02	2,4,8,12,14
T03	4,9,13
T04	2,4,8
T05	1,5,7
T06	8,9,10,11,12
T07	3,4,9,13,15
T08	8,12,14
T09	5,6
T10	3
T11	8,12
T12	5,6,7,8,11,12
T13	1,2,3,4,6,7
T14	4,8,12
T15	3,4,6
T16	4,6,7
T17	7,8
T18	3,4,13,15
T19	3,5,9
T20	3,5,7,9
T21	1,3,5,7,9
T22	2,4,12,14
T23	9
T24	4,9,13,15
T25	4,9,15
T26	4,14
T27	12
T28	4,9,13
T29	4,12,14
T30	5,9
T31	1,5,7,9
T32	13
T33	2,8,14
T34	3,6,7
T35	9,13
T36	9,13,15
T37	5,7
T38	2,4,14
T39	10,12
T40	8,9
T41	2,3,4,6
T42	8,9
T43	3,4,9,13
T44	8,11,12
T45	9,11,12
T46	3,5,7,9
T47	3,9
T48	8,10,11,12
T49	8,11,12
T50	1,3,5,7

Table 9: Clusters with transactions.

Cluster No.	Transactions
Cluster 0	T12,T44,T48,T49,T06
Cluster 1	T13,T41,T50,T01,T15,T16,T21,T34
Cluster 2	T01,T21,T31,T46,T50,T20,T05,T13,T19
Cluster 3	T02,T22,T29,T33,T38,T04,T08,T14
Cluster 4	T06,T48,T49,T12,T44,T45
Cluster 5	T07,T18,T24,T43,T03,T25,T28,T36
Cluster 6	T21,T01,T31,T46,T50,T20,T05,T13,T19

Cluster 7	T22,T02,T29,T38
Cluster8	T24,T07,T18,T03,T25,T28,T36,T43

Above table 9 shows the nine clusters. In this transaction which has more number of items kept at first position and rest all transaction in cluster with matching of minimum similarity items 3 or more items, with first transaction put in to same cluster.

For example in cluster 0 , in which T12 has highest number of items i.e. six items <5,6,7,8,11,12> and all other transaction (T44,T48,T49,T06) in cluster 0 contains the items with minimum 3 similar items hence put it on cluster 0. Based on this concept all other clusters are generated.

After generating clusters apply matrix algorithm and it will generate matrix (Table 10) for each cluster as well as generate FIS (frequent item set) from each cluster as per matrix algorithm. after generating FIS from all cluster , put it on single list.

Table 10: Matrix of all clusters with frequent item set

Cluster 0

	T12	T44	T48	T49	T06
5	1	0	0	0	0
6	1	0	0	0	0
7	1	0	0	0	0
8	1	1	1	1	1
11	1	1	1	1	1
12	1	1	1	1	1

FIS: <8,11,12>

Cluster 1

	T13	T41	T50	T01	T15	T16	T21	T34
1	1	0	1	1	0	0	1	0
2	1	1	0	0	0	0	0	0
3	1	1	1	1	1	0	1	1
4	1	1	0	0	1	1	0	0
6	1	1	0	0	1	1	0	1
7	1	0	1	1	0	1	1	1

FIS: <1,3,7> <3,4,6>

Cluster 2

	T01	T21	T31	T46	T50	T20	T05	T13	T19
1	1	1	1	0	1	0	1	1	0
3	1	1	0	1	1	1	0	1	1
5	1	1	1	1	1	1	1	0	1
7	1	1	1	1	1	1	1	1	0
9	1	1	1	1	0	1	0	0	1

FIS: <1,5,7,9> <1,3,5,7> <3,5,7,9>

Cluster 3

	T02	T22	T29	T33	T38	T04	T08	T14
2	1	1	0	1	1	1	0	0
4	1	1	1	0	1	1	0	1
8	1	0	0	1	0	1	1	1
12	1	1	1	0	0	0	1	1
14	1	1	1	1	1	0	1	0

FIS: <4,12,14> <2,4,14>

Cluster 4

	T06	T48	T49	T12	T44	T45
8	1	1	1	1	1	0
9	1	0	0	0	0	1
10	1	1	0	0	0	0
11	1	1	1	1	1	1
12	1	1	1	1	1	1

FIS: <8,11,12>

Cluster 5

	T07	T18	T24	T43	T03	T25	T28	T36
3	1	1	0	1	0	0	0	0
4	1	1	1	1	1	1	1	0
9	1	0	1	1	1	1	1	1
13	1	1	1	1	1	0	1	1
15	1	1	1	0	0	1	0	1

FIS: <4,9,13> <9,13,15>

Cluster 6

	T21	T01	T31	T46	T50	T20	T05	T13	T19
1	1	1	1	0	1	0	1	1	0
3	1	1	0	1	1	1	0	1	1
5	1	1	1	1	1	1	1	0	1
7	1	1	1	1	1	1	1	1	0
9	1	1	1	1	0	1	0	0	1

FIS: <1,3,5,7> <3,5,7,9> <1,5,7,9>

Cluster 7

	T02	T22	T29	T38
2	1	1	0	1
4	1	1	1	1
8	1	0	0	0
12	1	1	1	0
14	1	1	1	1

FIS: <2,4,14> <4,12,14>

Cluster 8

	T24	T07	T18	T03	T25	T28	T36	T43
4	1	1	1	1	1	1	0	1
9	1	1	0	1	1	1	1	1
13	1	1	1	1	0	1	1	1
15	1	1	1	0	1	0	1	0

FIS <4,13,15> <9,13,15> <4,9,13>

Below figure 11 Shows the snap shot and implementation of proposed algorithm. where it will be taking Number of cluster and threshold value as input and generating output by generating all cluster as well as matrix of each cluster and frequent item set of all cluster after finishing the matrix algorithm as well as it show time required to generate frequent item sets.

Figure 2: Snap of proposed algorithm with implementation

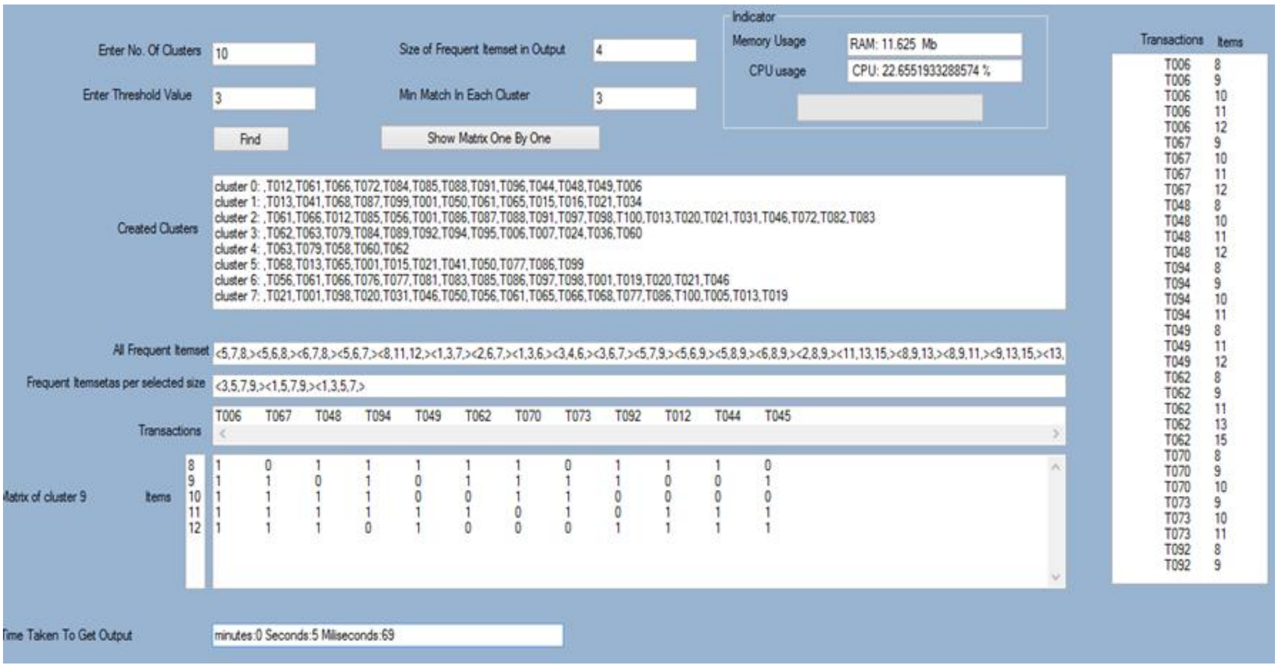
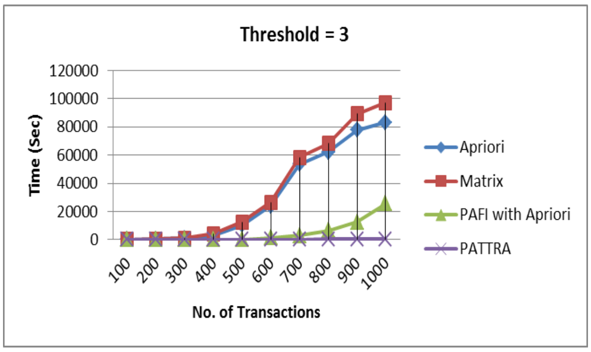


Figure 3: Time required by different algorithms on different set of transactions.



4.1 Statistical analysis

The experiment is conducted on dataset, which composed of 1000 transactions and average size of transaction is 5 items and based on that performance is measured with different parameters.

The performance measured on different set of transaction with fixed threshold =3 is shown in Table 12 and figure 3.It shows that matrix and apriori algorithm is required more time when transaction size is increased as compared to PAFI with apriori and PATTRA.

Table 12: Time required to generate frequent item set with threshold = 3 on different algorithm.

No. of Transaction	Apriori (in Sec)	Matrix (in Sec)	PAFI with Apriori (in Sec)	PATTRA (in Sec)
100	10	23	7	5
200	456	302	16	9
300	992	1036	28	15
400	3189	4120	97	41
500	10349	12657	239	74
600	23890	26534	1253	158
700	58672	57249	2802	221
800	70213	68126	6544	307
900	87890	89343	12472	416
1000	93245	97128	25513	562

The performance measured on fixe dataset and with different threshold is shown in Table 13 and figure 4. It also shows that matrix and apriori algorithm is required more time when threshold is decreased as compared to PAFI with apriori and PATTRA.

Table 13: Time required finding out frequent item set with 1000 transactions.

Threshold	Apriori (in sec)	Matrix (in sec)	PAFI with Apriori (in sec)	PATTRA (in sec)
3	93245	97128	25513	562
4	64345	76345	2702	559
5	45246	54389	961	485
6	20367	39455	705	482

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
International Conference & workshop on Advanced Computing 2014 (ICWAC 2014) – www.ijais.org

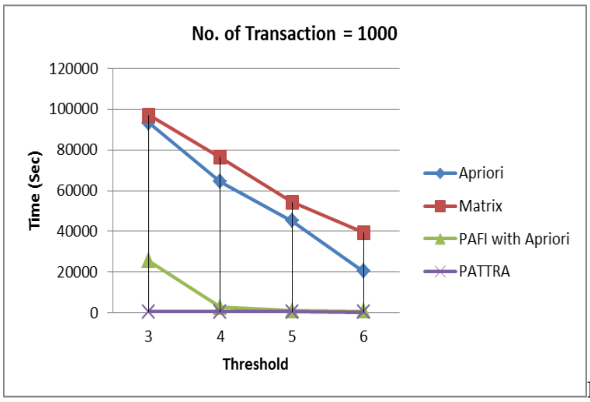


Figure 4: Time required by different algorithms with different threshold.

PATTRA as well as PAFI with apriori both algorithm uses clustering technique hence the performances is measured on both algorithms is shown in Table 14 and figure 5, 6, 7 with different thresholds and with different size of data set. It shows that PAFI with apriori gives faster FIS when number of transaction less than 500 and threshold =5 but when the transaction increases it becomes slower than PATTRA algorithm.

Table 14: Time (in second) required to generate frequent item set with threshold = 3, 4, 5 with different set of transactions.

No. of Transaction	No. of cluster	Threshold	PAFI with Apriori (in sec)	PATTRA (in sec)
100	10	3	7	6
		4	4	5
		5	2	5
200	20	3	16	9
		4	10	9
		5	4	8
300	30	3	28	15
		4	14	15
		5	9	14
400	40	3	97	41
		4	37	41
		5	22	40
500	50	3	239	74
		4	85	73
		5	48	72
600	60	3	1253	158
		4	376	157
		5	159	156
700	70	3	2802	221
		4	478	219
		5	292	218
800	80	3	6544	307
		4	1332	306
		5	423	300
900	90	3	12472	416
		4	1749	412
		5	608	414
1000	100	3	25513	562
		4	2702	559
		5	961	485

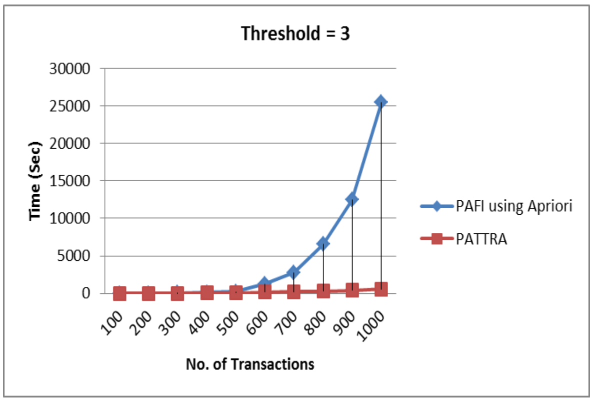


Figure 5: Time required by PATTRA and PAFI with apriori when threshold =3.

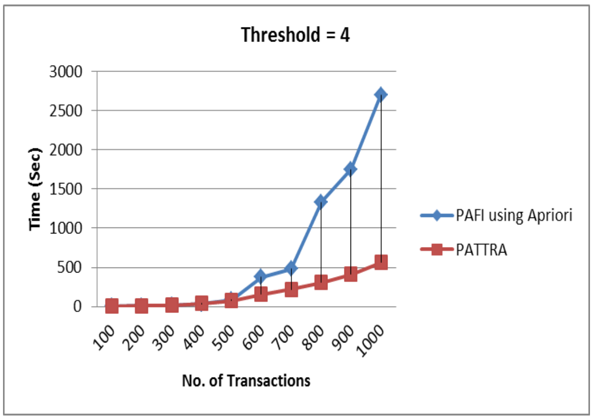


Figure 6: Time required by PATTRA and PAFI with apriori when threshold =4.

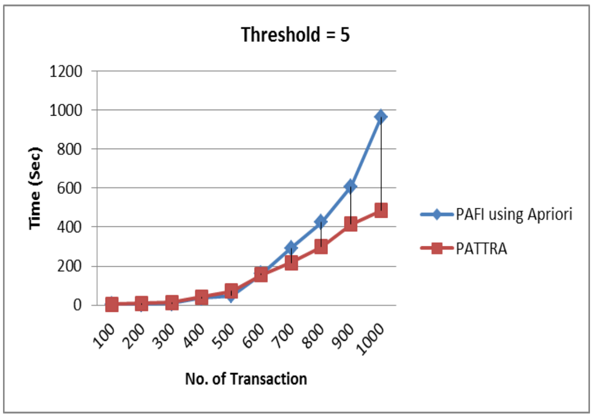


Figure 7: Time required by PATTRA and PAFI with apriori when threshold =5.

When Number of transactions is less than 500 and threshold is 6 than PAFI with apriori work faster than PATTRA but as the threshold value decreases and number of transaction increases PATTRA is faster than PAFI with apriori.

There are few constraints in PATTRA algorithm as follow:

1. In PATTRA, database is divided in to how many clusters is decided by user because of that time required to find out FIS varies depending on number of clusters as well as if less number of clusters is selected less than required clusters than it may drop some of the FIS.

2. Minimum how many matching items should be placed into the cluster with top most transaction of cluster is also decided by user. For e.g. if minimum match items = 3 than PATTRA will generate FIS of more than or equal to 3 frequent items groups and depends on that also time may change.

5. CONCLUSION

In this paper, the novel algorithm PATTRA is proposed where the entire database divided into partitions of variable sizes, each partition will be called a cluster than each cluster is converted into matrix by matrix algorithm and generate frequent item set from each cluster. Here Instead of entire database only each cluster is considered one at a time hence time required to swap in and swap out from memory is less compare to apriori and Matrix algorithm as well as computational speed will be increase. It also reduces the redundant database scan and improves the efficiency.

Performance studies shows that PATTRA take 50% to 80% less time than PAFI with apriori algorithm to generate FIS as well as if threshold value changes on same dataset than also PATTRA take almost same amount of time whereas existing algorithm varies with respect to change in threshold value. It also shows that Matrix and apriori is not efficient for large dataset.

Hence novel algorithm PATTRA gives better performance than existing algorithms when there is large dataset and it gives better time complexity and space complexity.

6. REFERENCES

[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