IMPROVING APRIORI ALGORITHM USING PAFI AND TDFI

Manali Patekar[1], Chirag Pujari[2], Juee Save[3]

[1,2,3]Computer Engineering, St. John College of Engineering And Technology,
Palghar Mumbai, (India)

ABSTRACT

Mining of Association rules in huge database is the challenging and tough task. To perform this Association an Apriori algorithm is mostly used. The work of Apriori algorithm is to find out the frequent item sets from large database. But using Apriori algorithm for finding frequent item has some limitations. It generates overfull candidates of frequent item sets from transactions,the algorithm needs to scan database again and again while finding frequent item sets from transaction. It will be not suitable for large database. It also requires more I/O load while accessing the database frequently. To solve the problems of Apriori algorithm, PAFI and TDFI method used in system. PAFI- An efficient Partition Algorithm for Mining Frequent Item sets and TDFI-Two Dimensional Approach for Mining Frequent Item sets. This algorithm divides transactions from the database into various partitions for finding frequent item sets. Then for each partition it finds the frequent item sets using a two dimensional approach which f reduces the number of scans in the database. So that the algorithms, PAFI and TDFI improve the efficiency

I. INTRODUCTION

Association rule is used for mining the data .This is one of the most important technique for mining. Association rule is used for mining where large database is stored like marketing, advertising and inventory control .This rule shows the relationships .The Relation or the Association between the data are mostly complicated .Sometimes data is hidden. The Apriori algorithm is used to find out the hidden items or data. Apriori algorithm is most important type of Association rule and it is very popular. But Apriori algorithm has two disadvantages: 1.It requires large IO load for scanning database again and again. 2. It also produces overfull candidates of frequent item sets which is generated in each scan. In this paper, we represent PAFI and TDFI algorithms to solve problems of Apriori algorithm, PAFI stands for Partition Algorithm for Mining Frequent Itemsets which is used to create clusters of similar data items .TDFI stands for Two Dimensional Approach Algorithm for Mining Frequent Itemsets which is used to find the frequent itemsets from each partition.

II. LITERATURE SURVEY

2.1 Research on Frequent Item sets Mining Algorithm based on Relational Database:

Mining association rules between large items is an important research direction of data mining, and the RDBMS is the most databases, so mining association rules in the relational database is a very important research direction.

In this paper, the frequent item sets mining algorithm based on relational database based on the study of those important mining association rule algorithm and the storage characteristics of the transaction set and items in the relational database, and present its concrete optimization and implementation method. This algorithm combines items in a transaction to generate item sets and counts the same item sets in all transactions, which improve the efficiency of execution. Moreover, this algorithm doesn't produce candidate item sets, and only scans transaction database once, so promotes considerable efficiency. The result of experiment show that, the frequent item sets mining algorithm based on RDBMS that has higher efficiency than the classical Apriori algorithm.

2.2. Performance Oriended Partition Algorithm for Minning Frequent Itemset:

In this paper, the Apriori Algorithm is the most well known association rule algorithm.It uses the largest itemset property . The subset of a large itemset must be large.The basic idea of Apriori algorithm is to generate item sets of a particular size and then scans the database to count these to see if they are large. Only those candidates that are large are used to generate candidates for the next database scan. Li is used to generate next Ci+1. L represent Large Item-set. C represents candidate items. All singleton item sets are used as candidates in the first pass. The set of large item sets of the previous scan, Li-1 is joined with itself to determine the candidates. Individual item sets must have all but one item in common in order to be combined.

2.3. Novel Approach to Improve Apriori Algorithm using Transaction Reduction and Clustering Algorithm based on matrix:

In this, An improved Apriori algorithm based on the matrix. To solve the bottleneck of the Apriori algorithm, we introduce an improved algorithm based on the matrix. the matrix is effectively use to indicate the affairs in the database and to deal with the matrix it uses the "AND operation" to produce the largest frequent item sets and others. The algorithm don't scan database frequently which is based on matrix , which reduce the spending of I/O. The new algorithm is good than the Apriori in the time complexity. it is not suitable for large database. PAFI algorithm is better for partitioning large database and so that partition each cluster or partition easily swap in or swap out .As well as Matrix method is better because from each cluster finding out frequent item set with less span of time. So that we using mixture of PAFI and Matrix based algorithm.

III. PROBLEM STATEMENT

Apriori algorithm is used to generate item sets of a particular size and then scan the database to count these to see if they are large. Only those number of candidates that are large are used to generate candidates for the next scan.Problem in Apriori algorithm is it may produce overfull candidates of frequent item sets. It needs great I/O load when frequently scans database. it scan database again and again and hence generate large no of candidate sets so that more memory is required.

IV. IMPLEMENTATION METHODOLOGY

4.1 PAFI (Partition Algorithm for Frequent Itemsets) ALGORITHM

Input:

1.Database D

2.Number of partitions P

```
Output:
Partitions with λ transactions
Begin
Number of transactions in each partition (λ)= Total
transactions in D/P //P<m is the random natural
number
FOR each partition Pi DO BEGIN
Take λ transactions in Pi
Put each ti in Pi
END
Return partitions with λ transactions.
```

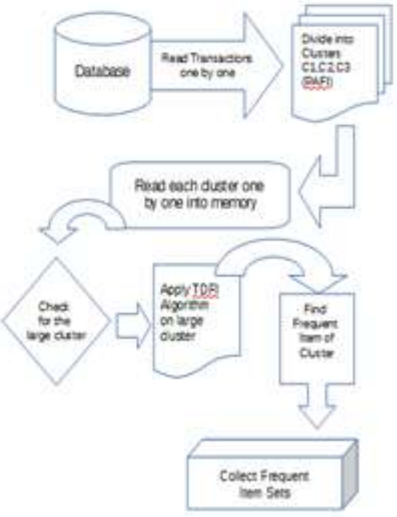
P6

4.2 TDFI (Two Dimensional Approach for Mining Frequent Itemset)

```
ALGORITHM:
Input:
1.Database D
2.Minimum support count min_sup
Output:
Frequent Itemsets
//Algorithm to find frequent itemset
FOR i= 1 to P DO BEGIN
FOR each partition Pi DO BEGIN
FOR each transaction t ∈ Pi DO BEGIN
Create a new row with the different number of items in D and mark it as „t‟ if purchased „f‟ if not purchased
FOR each item Ii ∈ Pi DO BEGIN
// for singleton itemsets Find supcount
END
If supcount <min_sup then delete $I_i$ from $P_i$;
FOR each item $I_i, I_j \in P_i$ in t DO BEGIN
// for 2 itemsets
Perform AND operations
Find supcount
END
If supcount <min_sup then delete $I_iI_j$ from $P_i$;
END
Repeat for n itemsets Until frequent itemsets occurs
END
```

```
L = L U L_i;
END
Return L; //L gives the set of all frequent itemsets
```

V. BLOCK DIAGRAM


Database
Read Transactions one by one
Divide into Clusters C1,C2,C3 (PAFI)
Read each cluster one by one into memory
Check for the large cluster
Apply TDFI Algorithm on large cluster
Find Frequent Item of Cluster
Collect Frequent Item Sets

VI. EXPERIMENTAL RESULTS:

Consider the following example having transactions in database.

On the given transaction, first we are performing PAFI which performs partitioning of database then for finding frequent itemsets from transaction we are applying two dimentional algorithm TDFI.

Steps:

1. Perform Partition algorithm(PAFI) on given set of transaction in database. Each partition having λ transaction. In this example we are getting p1 and p2 this two partitions.

2. Create a new row with the different number of items for each partition and for each transaction in D and if the item is present mark it as “t‘‘ otherwise mark it as "f".

3. Take minimum support count. E.g.: min_sup=2

4. For all items in L1, calculate the support count.

5. Delete those item whose support count is less than min_sup.

6.To calculate 2-itemsets, perform AND operation. Then again calculate support count for all items in L2 and delete the items whose support count is less then min_sup from L2.

7. Repeat the above steps for 3-itemsets. Repeat until we get the frequent itemsets.

D

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

Large itemset-1 L1:

P1 p2

TID	ITEMS
T1	A,B,E
T2	B,D
T3	B,C
T4	A,B,D
T5	A,C

TID	ITEMS
T6	B,C
T7	A,C
T8	A,B,C,E
T9	A,B,C

Take Min_sup=2

	A	B	C	D	E
T1	T	t	f	f	t
T2	F	t	f	t	f
T3	F	t	t	f	f
T4	T	t	f	t	f
T5	T	f	t	f	f
	3	4	2	2	1

Large itemset-2 L2

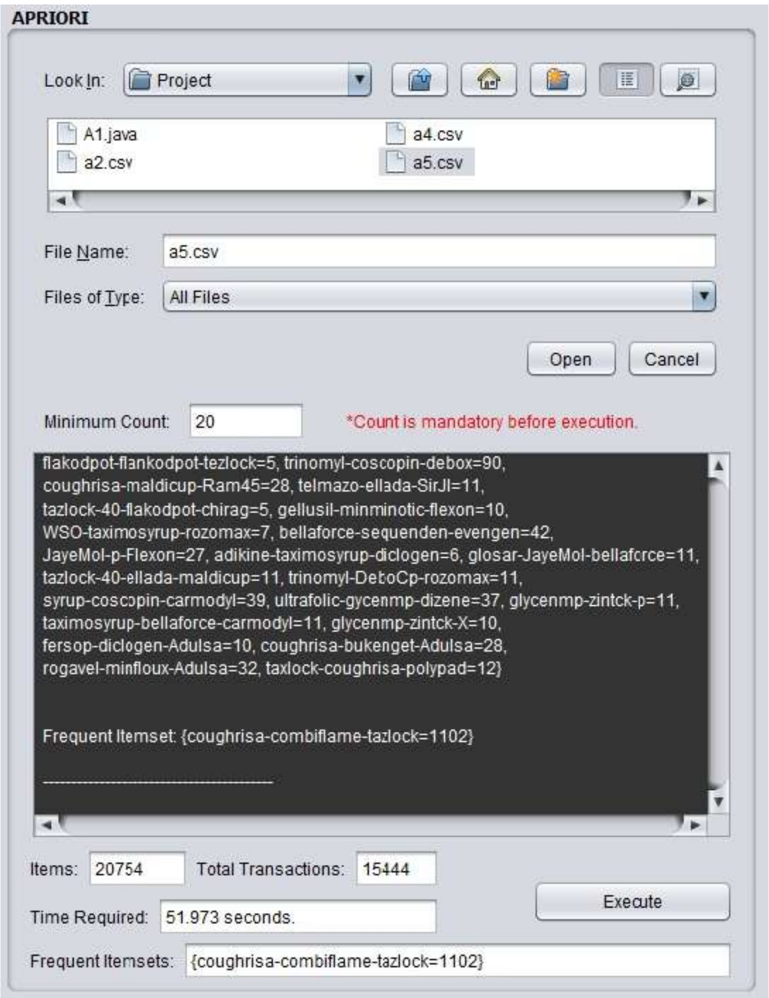
	AB	AC	AD	BC	BD	CD
T1	t	f	f	f	f	f
T2	f	F	f	f	t	f
T3	f	F	f	t	f	f
T4	t	F	t	f	t	f
T5	f	T	f	f	f	f
	2	1	1	1	2	0

Large itemset-3 L3

	ABD
T1	F
T2	F
T3	F
T4	T
T5	F
	1

Frequent itemset=AB and BD

VII. OUTPUT



Look In: Project

File Name: a5.csv

Files of Type: All Files

Minimum Count: 20

*Count is mandatory before execution.

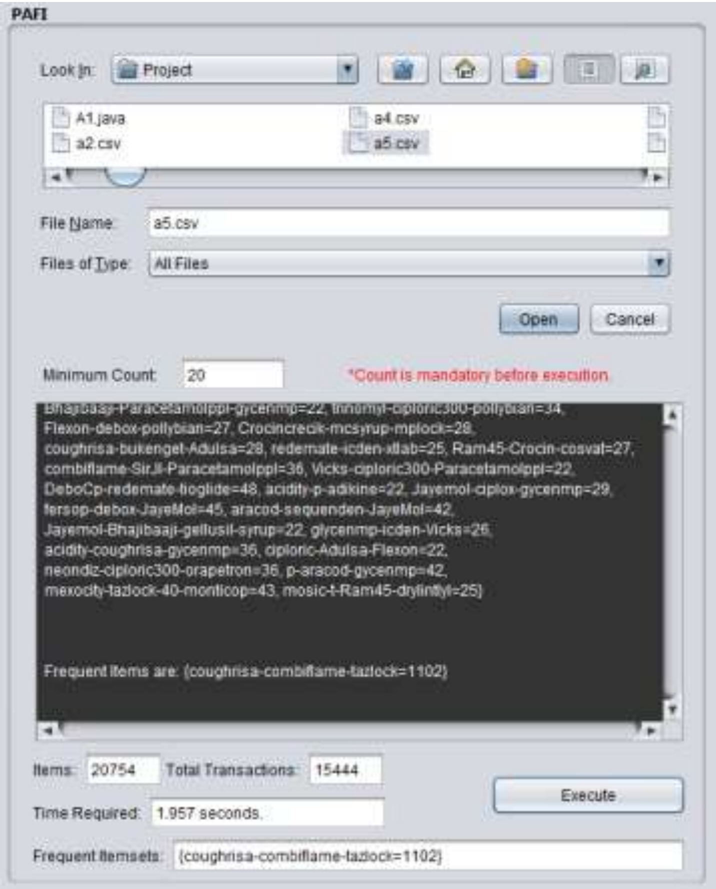
```
flakodpot-flankodpot-tezlock=5, trinomyl-coscopin-debox=90,
coughrisa-maldicup-Ram45=28, telmazo-ellada-SirJI=11,
tazlock-40-flakodpot-chirag=5, gellusil-minminotic-flexon=10,
WSO-taximosyrup-rozomax=7, bellaforce-sequenden-evengen=42,
JayeMol-p-Flexon=27, adikine-taximosyrup-diclogen=6, glosar-JayeMol-bellafcrce=11,
tazlock-40-ellada-maldicup=11, trinomyl-DeboCp-rozomax=11,
syrup-coscopin-carmodyl=39, ultrafolic-gycenmp-dizene=37, glycenmp-zintck-p=11,
taximosyrup-bellaforce-carmodyl=11, glycenmp-zintck-X=10,
fersop-diclogen-Adulsa=10, coughrisa-bukenget-Adulsa=28,
rogavel-minfloux-Adulsa=32, taxlock-coughrisa-polypad=12}


Frequent Itemset: {coughrisa-combiflame-tazlock=1102}

-----------------------------------------
```

Items: 20754 Total Transactions: 15444

Time Required: 51.973 seconds.

Frequent Itemsets: {coughrisa-combiflame-tazlock=1102}

Fig. 1. Apriori Algorithm



File Name: a5.csv

Files of Type: All Files

Minimum Count: 20

*Count is mandatory before execution

```
Bhajibaaji-Paracetamolppl-gycenmp=22, Innomyl-ciploric300-pollybian=34,
Flexon-debox-pollybian=27, Crocincrecik-mcsyrup-mplock=28,
coughrisa-bukenget-Adulsa=28, redemate-icden-xllab=25, Ram45-Crocin-cosval=27,
combiflame-SirJl-Paracetamolppl=36, Vicks-ciploric300-Paracetamolppl=22,
DeboCp-redemate-tioglide=48, acidity-p-adikine=22, Jayemol-ciplox-gycenmp=29,
fersop-debox-JayeMol=45, aracod-sequenden-JayeMol=42,
Jayemol-Bhajibaaji-gellusil-syrup=22, glycenmp-icden-Vicks=26,
acidity-coughrisa-gycenmp=36, ciploric-Adulsa-Flexon=22,
neondiz-ciploric300-orapetron=36, p-aracod-gycenmp=42,
mexocity-tazlock-40-monticop=43, mosic-t-Ram45-drylintyl=25}


Frequent Items are: {coughrisa-combiflame-tazlock=1102}
```

Items: 20754 Total Transactions: 15444

Time Required: 1.957 seconds.

Frequent Itemsets: {coughrisa-combiflame-tazlock=1102}

Fig. 2. PAFI with TDFI

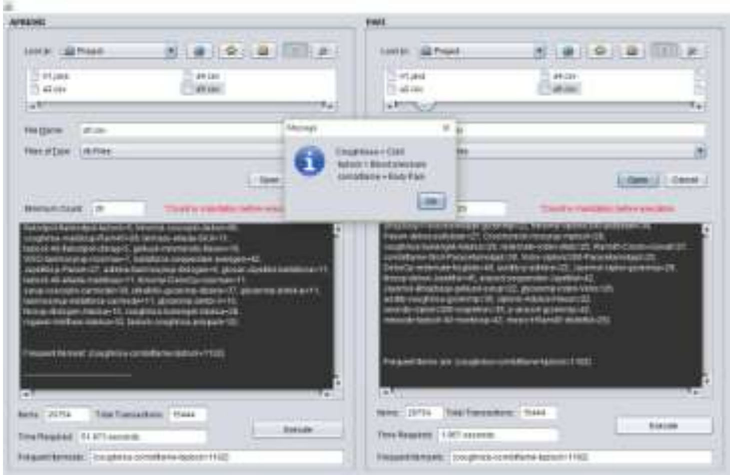
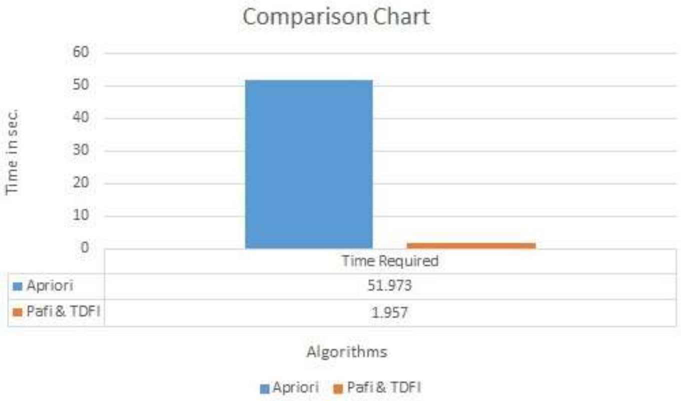


Fig.3 Frequent Itemset Generator

- ➢ Finding Frequent Item sets using medical dataset.
- ➢ For No. of transactions and Items check the screenshots provided above.
- ➢ Finding frequent medicines sold in different areas and on basis of medicines identified, selected dataset of particular area is prone to which type of sickness or disease.



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"*

[4]. Zhu Yixia, Yao Liwen, Huang Shuiyuan, Huang Longjun, *" A matrix and trees". association rules* Computer science. 2006, 33(7):196-198*mining algorithm*

[5]. Arun K Pujari. Data Mining Techniques (Edition 5):Hyderabad, India: Universities Press (India) Private Limited, 2003.

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

[7]. Jiawei Han. Data Mining, concepts and Techniques: San Francisco, CA: Morgan Kaufmann Publishers.,2004.

[8]. Akhilesh Tiwari, Rajendra K. Gupta, and Dev Pra-kash Agrawal "Cluster Based Partition A pproach for Mining Frequent Itemsets" In Proceedings of the IJCSNS International Journal of computer Science and Network Security, VOL.9 No.6, June 2009.

[9]. R.K. Gupta. Development of Algorithms for New Association Rule Mining System, Ph.D. Thesis, Submitted to ABV-Indian Institute of information Technology & Management, Gwalior, India, 2004.

[10]. M. Houtsma and A. Swami. Set Oriented Mining for Association Rules in Relational Databases. In Proceedings of 11th International conference on Data Engi-neering, 1995, pp 25-33 .