ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

AN EFFICIENT APPROACH FOR THE IMPLEMENTATION OF FP TREE

SADHANA KODALI[1], KAMALAKAR M[2], CH. GAYATRI [3], K.PRAVALLIKA[4]

Assistant professor, Department of IT, Lendi institute of Engineering and Technology, Vizianagaram, AndhraPradesh,India.[1]

Assistant Professor, Department of CSE, Raghu Engineering College, Visakhapatnam, Andhra Pradesh,India[2]

10KD1A1212, IV/IV B Tech, Department of IT, Lendi institute of Engineering and Technology, Andhra Pradesh, India.[3]

10KD1A1239, IV/IV B Tech, Department of IT, Lendi institute of Engineering and Technology, Andhra Pradesh, India.[4]

***ABSTRACT**:* Frequent pattern mining is one of the most common mining techniques to identify the frequent patterns in large data sets. The Apriori algorithm is one algorithm very efficient for mining frequent patterns. But the drawback is it generates a number of candidate item sets. The FP tree is a frequent pattern technique without candidate item set generation. We have proposed an approach for improving the performance of the FP tree using the parameter average support count.

KEYWORDS*:* Frequent pattern mining, Apriori algorithm, FP tree, candidate item sets, average support count.

I. INTRODUCTION

Mining frequent patterns in transactional databases is a popular field of study in data mining. Apriori algorithm [1] is a traditional algorithm for finding the frequent patterns using some statistical measures such as support count and confidence. But the drawback of the Apriori algorithm is repeated database scanning and pruning of infrequent candidate item sets. Generating of the candidate item sets and pruning of the infrequent items is cost effective if large data sets are considered. An approach was proposed for mining frequent patterns without candidate generation [5] and an efficient FP-tree based mining method, FP-growth was developed, for mining the complete set of frequent patterns by pattern fragment growth. [2].In our paper we would like to improve the efficiency of this algorithm by proposing an approach to still reduce the infrequent candidate generation and improve the space and time complexity of the algorithm. The algorithm *efficient FP tree has* the following major steps:

-Scan the database for the first pass and prune the item sets with infrequency using a threshold value *avgsup*. [3].
-Now the large database is condensed to a smaller data structure called the FP tree.
-Partitioning-based, divide-and-conquer method is used to decompose the mining task into a set of smaller tasks for mining confined patterns in conditional databases, which dramatically reduces the search space.

II. PROCESSING THE DATABASE

The first step is to preprocess the database before applying the algorithm. We take an example dataset to which we apply the formulas derived in [3].We compare this with the normal way of generating the fp tree and show the difference

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

between these two approaches. We proposed a measure called as *avgsup* which is obtained by calculating the total support counts of n items and calculating the average of these items. This approach is much better for pruning of the infrequent items. The reason for this is, all the infrequent items fall below this threshold value called the *avgsup*. If there are {x1, x2, x3…. xk} items the *avgsup* is calculated using [3]

$$\text{AvgSup}(x) = \frac{\sum_{n=1}^{K} \text{Sup}(X\,k)}{K}$$

The example dataset is taken in the following table:

Table: 1

T1	c , e
T2	a, b , c, e
T3	b,c,d,e
T4	b,c
T5	a,b
T6	b,f
T7	b
T8	a,c,e
T9	a,c,d

The individual support counts of the items in the transaction are given in Table 2:

a	4
b	6
c	6
d	2
e	4
f	1

Step 1: Calculation of average support:

$$\overline{\text{AvgSup}(x)} = \frac{\sum_{n=1}^{k} \text{Sup}(X\,k)}{K}$$

Avgsup= (4+6+6+2+4+1)/6

The support count of each item divided by the number of items. The value of *avgsup* for our example would be 3.5 for which we consider the floor value. The floor value of the *avgsup* denoted as $\lfloor avgsup \rfloor$ is given as 3.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

Step-2: Pruning of the transactional database with *avgsup* count less than 3.We get the following items considered as the most frequent items

Table 3a: Transactional data set after pruning with avgsup.

T1	c , e
T2	a, b , c, e
T4	b,c
T5	a,b
T7	b
T8	a,c,e

After removing the list of transactions which do not satisfy the threshold *avgsup* we get the above table.

Table 3b: Table which shows items with support counts

b	*4*
c	4
a	3
e	2

III. CONSTRUCTION OF FP-TREE

An fp tree is constructed for the items in Table 3 using the normal approach for the construction of fp tree.

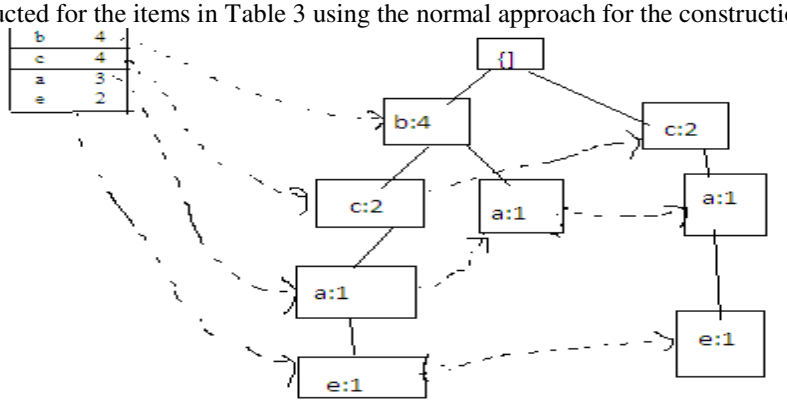


Figure: 1 Fp tree after applying avgsup

The same data is used to normally generate the fp tree. Figure: 2 represents the normal way of construction of the fp tree [6].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

TID	Itemset
100	c, e
200	a, b, c, e
300	b, c, d, e
400	b, c
500	a, b
600	b, f
700	b
800	a, c, e
900	a, c, d

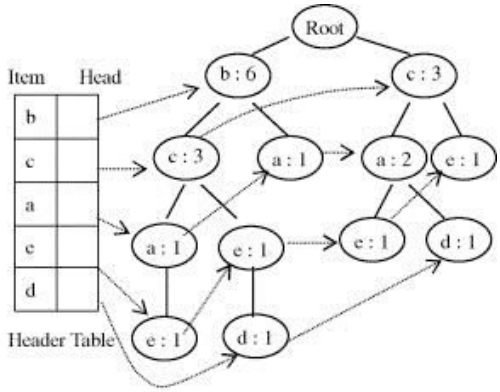


Figure-2: fp tree.

Observing the tree in figure 1 we can say it is an optimal way of implementing the fp tree. The tree in figure 1 clearly reduces the space complexity when compared with the normal way of generating the fp tree.

Graph 1: Graph which shows the items below the avgsup value.

From this we can understand that items d and f are infrequent and those item-sets with the infrequent items become less frequent and the elimination of such items from the transactional database for the construction of the fp tree will not affect the frequent pattern.

IV EFFICIENCY OF THE PROPOSED APPROACH

Using the same transactional database we apply Apriori algorithm and compare our results of the efficient fp tree with that of the existing Apriori algorithm.The apriori algorithm follows the steps below for the generation of frequent item-sets.

Step1: Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k-1.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

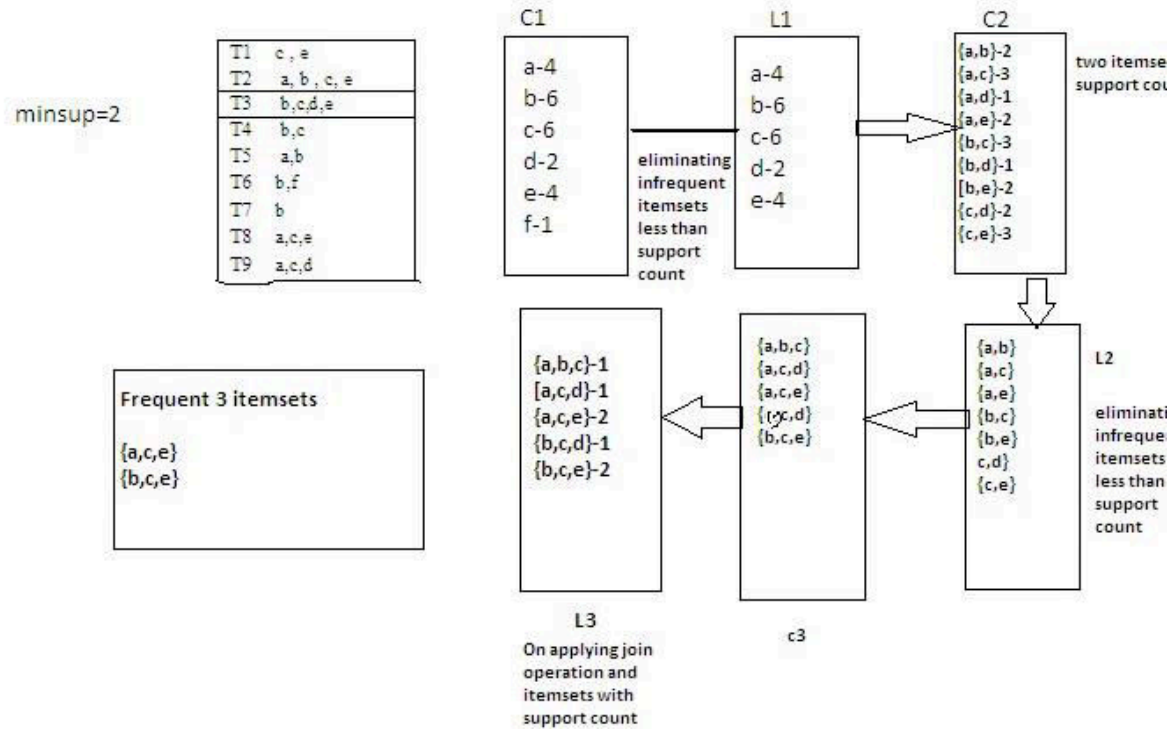
(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

Step2: Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. [4]

Step3: After that, it scans the transaction database to determine frequent item sets among the candidates.

Figure-3: Applying apriori for the transactional database.



Observing the figure 1 and figure 3, we have the same frequent 3 item-sets. In our proposed approach efficient *fp tree* we obtained the frequent 3-item-sets only with one scan. But with the existing technique Apriori algorithm we need to scan the database four to five times and apply the pruning step to eliminate the infrequent item sets.

V CONCLUSION

From the above we can observe that the efficiency of the proposed approach is more when compared with the existing techniques. The approach proposed by us will improve the time and space complexity. The approach proposed compresses the dataset and requires only one scan and two passes for the identification of frequent itemsets.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

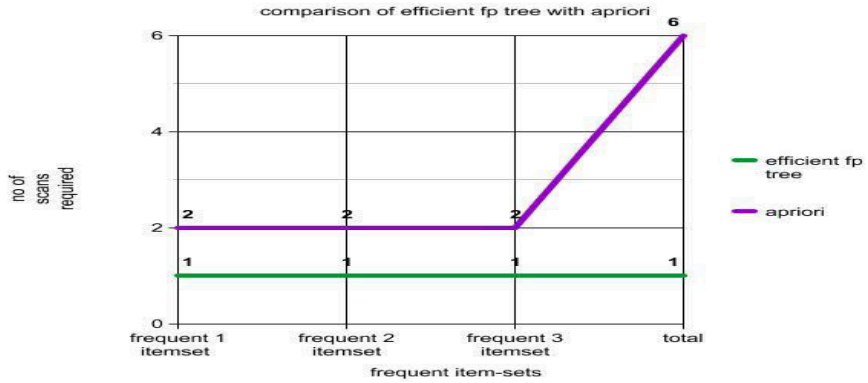
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

When we compare the two approaches we get the graph as follows:

Graph-2: Comparison of Apriori with efficient FP tree



REFERENCES

[1] R. Agrawal and R. Srikant. "Fast algorithms for mining association rules in large databases." Re-search Report RJ 9839, IBM Almaden Research Center, San Jose, California, June 1994.
[2] JIAWEI HAN, JIAN PEI, YIWEN YIN, RUNYING MAO **"**Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach "
Proceedings of Data Mining and Knowledge Discovery, 8, 53–87, 2004 Kluwer Academic Publishers. Manufactured in The Netherlands.
[3]Sadhana kodali,Kamalakar M ," An Approach for Improving the Performance of the Apriori Algorithm published in IIJDWM" volume 3 issue 2.May 2013.
[4] en.wikipedia.org/wiki/Apriori_algorithm
[5] Mining Frequent Patterns without Candidate Generation proceedings of SIGDOM '2000. Jiawei Han, Jian Pei, and Yiwen Yin.
[6] Christian Borgelt "An Implementation of the FPgrowth Algorithm " OSDM '05 Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations.

BIOGRAPHY

Mrs SADHANA KODALI. M TECH Working as an assistant professor in Lendi Institute of Engineering and technology.Previously worked in Raghu Institute of technology and Koneru Lakshmaiah college of Engineering. Work experience:6.5 years. Publications: Published a paper in IFRSA International Journal of Data warehousing and mining, volume 3, issue 2.Published papers in 2 international Conferences. (ICDF 2008 and ICSTM 2013 conducted by WAIRCO)

MR KAMALAKAR MEDURI, Mtech.Working as an Assistant professor in Raghu Engineering College. Work Experience : 7 years. Publications: Published a paper in IFRSA International Journal of Data warehousing and mining, volume 3, issue 2.Published papers in 2 international Conferences. (ICDF 2008 and ICSTM 2013 conducted by WAIRCO).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

3. Gayatri Ch.(10KD1A1212) Final Year Btech student .Department of IT.Lendi Institute of Engineering and Technology.

4. Pravallika K.(10KD1A1239) Final Year Btech student. Department of IT.Lendi Institute of Engineering and Technology.