ISCA Journal of Engineering Sciences ______________________ *ISCA J. Engineering Sci.*
Vol. **1(1),** 14-18, July **(2012)**

A New Approach to Mine Frequent Itemsets

Patel Tushar S. and Amin Kiran R.
Computer Engineering Department, UVPCE, Kherva, Gujarat, INDIA

Available online at: www.isca.in
Received 30th May 2012, revised 1st June 2012, accepted 2nd June 2012

Abstract

Mining frequent patterns in transaction databases and many other kinds of databases has been studied popularly in data mining research. Methods for efficient mining of frequent itemsets have been studied extensively by many researchers. However, the previously proposed methods still encounter some performance bottlenecks when mining databases with different data characteristics. The time required for generating frequent itemsets plays an important role. And also the poor efficiency of counting candidate itemset's support count. In this study, we propose a new frequent itemsets tree (FI-tree) structure, which is used for storing frequent itemsets and their Tid sets. A distinct feature of this method is that it has runs fast in different data characteristics. Our study shows that a new approach has high performance in various kinds of data, outperforms the previously developed algorithms in different settings, and is highly scalable in mining different databases.

Keywords: Mining, frequent itemset, adult, hepatitis, heart.

Introduction

The term data mining or knowledge discovery in database has been adopted for a field of research dealing with the automatic discovery of implicit information or knowledge within the databases. The implicit information within databases, mainly the interesting association relationships among sets of objects that lead to association rules may disclose useful patterns for decision support, financial forecast, marketing policies, even medical diagnosis and many other applications.

Finding frequent item sets is one of the most investigated fields of data mining. The problem was first presented in paper mining association rules between sets of items in large databases by Agrawal[1]. To analyze the huge amount of data thereby exploiting the consumer behavior and make the correct decision leading to competitive edge over rivals[2]. Frequent itemsets are appear in a data set with frequency no less than a user-specified threshold. Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases such as association rules, correlations, sequences, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems. Also sequential association rule mining is one of the possible methods to analysis of data used by frequent itemsets[3].

Frequent pattern mining was first proposed by Agrawal et al.[1] form market basket analysis in the form of association rule mining. It analyses customer buying habits by finding associations between the different items that customers place in their "shopping baskets". For instance, if customers are buying milk, how likely are they going to also buy cereal (and what kind of cereal) on the same trip to the supermarket? Such information can lead to increased sales by helping retailers do selective marketing and arrange their shelf space.

This data mining task and its associated efficient mining algorithms, there have been hundreds of follow-up research publications, on various kinds of extensions and applications, ranging from scalable data mining methodologies, to handling a wide diversity of data types; various extended mining tasks, and a variety of new applications.

In this paper, we propose a new frequent itemsets tree (FI-tree) structure, which is used for storing frequent itemsets and their Tid sets. A distinct feature of this method is that it has runs fast in different data characteristics. Our study shows that a new approach has high performance in various kinds of data, outperforms the previously developed algorithms in different settings, and is highly scalable in mining different databases.

Need of frequent itemset mining: Studies of frequent itemset (or pattern) mining is acknowledged in the data mining field because of its broad applications in mining association rules, correlations, and graph pattern constraint based on frequent patterns, sequential patterns, and many other data mining tasks. Efficient algorithms for mining frequent itemsets are crucial for mining association rules as well as for many other data mining tasks. The major challenge found in frequent pattern mining is a large number of result patterns. As the minimum threshold becomes lower, an exponentially large number of itemsets are generated. Therefore, pruning unimportant patterns can be done effectively in mining process and that becomes one of the main topics in frequent pattern mining. Consequently, the main aim is to optimize the process of finding patterns which should be efficient, scalable and can detect the important patterns which can be used in various ways[4].

Related work: Several algorithms for mining associations have been proposed in the literature. Almost all the algorithms produce frequent itemsets on the basis of minimum support. Apriori algorithm[5] is quite successful for market based analysis in which transactions are large but frequent items generated is small in number. The Apriori variations (DHP, DIC, partition, and sample) algorithms among them DHP[6] tries to reduce candidate itemsets and others try to reduce database scan. DHP works well at early stages and performance deteriorates in later stages and also results in I/O overhead. For DIC[7], partition[8], sample algorithm[9] performs worse where database scan required is less then generating candidates For vertical layout based algorithm include Eclat[10] claims to be faster than Apriori but require larger memory space then horizontal layout based because they needs to load candidate, database and TID list in main memory. For projected layout based algorithms include FP-Tree[11] and H-mine[12], performs better then all discussed above because of no generation of candidate sets but the pointes needed to store in memory require large memory space. For SaM algorithm[13] are an exceptionally simple algorithm and data structure, but the points needed to store in memory require large memory space.

Methodology

In this section we describe our new approach for frequent itemset mining is described. This approach is based on a new frequent itemsets tree (FI-tree) structure, which is used for storing frequent itemsets and their Tid sets.

Now, steps of the proposed approach for mining frequent itemsets from different data characteristics: i. Assume that the frequent 1-itemsets and transaction sets, require no more memory that available and also there is space for generating candidate 2-itemsets from frequent 1-itemset. Scan database and find frequent 1-itemsets, at the same time obtain transaction sets, which includes the Itemset. ii. Generate candidate 2-itemsets from frequent 1-itemset only. iii. The candidate 2-itemsets whose node count is lower than min support using their FI-tree data structure, it will be pruned off. Now frequent Itemset tree contains only frequent 2-itemsets at the second level. iv. Consequently, for each frequent 3, 4… n–Itemset, scan the database to approve the consistence of the Itemset.

This is an example of mining frequent itemsets based on proposed new approach (figure 1). There are six transactions in this database, that is |D|=6.

Results and Discussion

Dataset: For the experiment we have used datasets of different application. These datasets was obtained from the UCI repository of machine learning databases[14]. The characteristics of the datasets selected for the experiment (table 1).

Table – 1
Details of dataset used in analysis

Files	Number of Records	Number of Columns
adult.D14.N48842.C2.num	48842	14
Hepatitis.D19.N155.C2.num	155	19
heart.D75.N303.C5.num	303	75

Result analysis: A detailed study to assess the performance of new approach with SaM algorithms. The performance metrics in the experiments is the total execution time taken and the support count for adult, hepatitis and heart datasets. For this comparison also same dataset were selected as for the above experiment with 30% to 70% of minimum support threshold. The execution time of new approach and SaM algorithms with different support threshold for adult data set (table 2).

Table – 2
Adult dataset execution time

Support	Total Time in Seconds	
	New Approach	SaM
30	8.12	9.85
40	5.69	6.72
50	3.56	4.51
60	1.99	2.69
70	1.01	1.7

The total execution time for the new approach and SaM algorithms large reduces with the increase in support threshold from 30% to 70% for adult dataset. The SaM algorithm takes more time as that compared to new approach (figure 2). The execution time of new approach and SaM algorithms with different support threshold for hepatitis data set (table 3).

Table – 3
Hepatitis dataset execution time

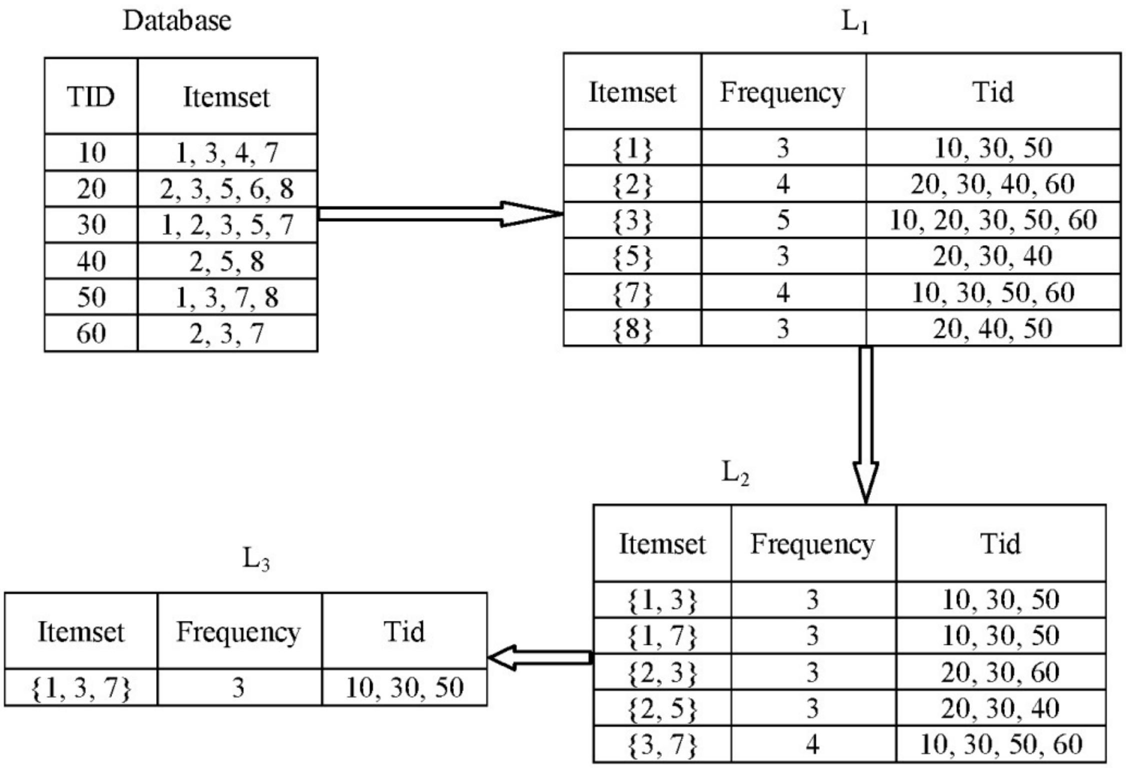
Support	Total Time in Seconds	
	New Approach	SaM
30	0.64	0.91
40	0.09	0.28
50	0.04	0.06
60	0.03	0.04
70	0	0

The total execution time for the new approach and SaM algorithms sharp decreases with the increase in support threshold from 30% to 40% for hepatitis dataset. The SaM algorithm takes more time as that compared to new approach (figure 3). The execution time of new approach and SaM algorithms with different support threshold for hepatitis data set (table 4).

Table – 4
Heart dataset execution time

Support	Total Time in Seconds	
	New Approach	SaM
30	0.05	0.07
40	0.04	0.06
50	0.03	0.05
60	0.02	0.03
70	0.01	0.02

The total execution time for the new approach and SaM algorithms small decreases with the increase in support threshold for heart dataset. The new approach takes less time compared to SaM (figure 4).



Database

TID	Itemset
10	1, 3, 4, 7
20	2, 3, 5, 6, 8
30	1, 2, 3, 5, 7
40	2, 5, 8
50	1, 3, 7, 8
60	2, 3, 7

L_1

Itemset	Frequency	Tid
{1}	3	10, 30, 50
{2}	4	20, 30, 40, 60
{3}	5	10, 20, 30, 50, 60
{5}	3	20, 30, 40
{7}	4	10, 30, 50, 60
{8}	3	20, 40, 50

L_2

Itemset	Frequency	Tid
{1, 3}	3	10, 30, 50
{1, 7}	3	10, 30, 50
{2, 3}	3	20, 30, 60
{2, 5}	3	20, 30, 40
{3, 7}	4	10, 30, 50, 60

L_3

Itemset	Frequency	Tid
{1, 3, 7}	3	10, 30, 50

Figure – 1
Mining frequent itemsets based on proposed new approach

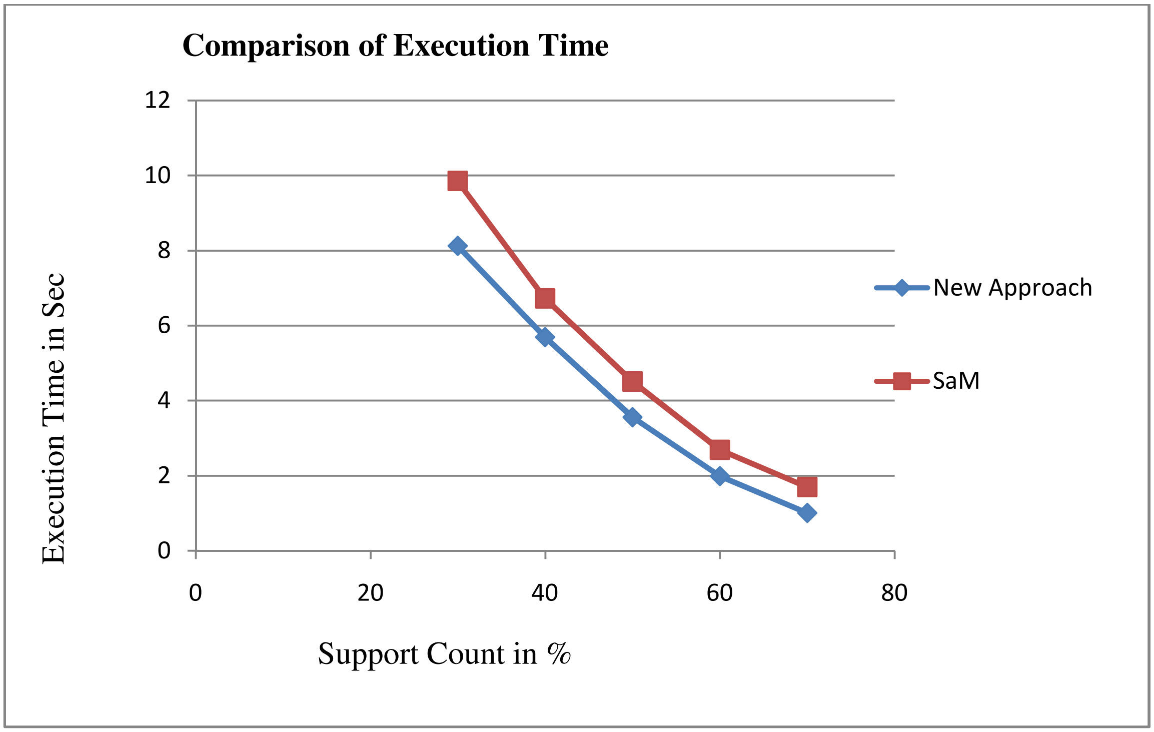


Figure – 2
Execution time for adult dataset

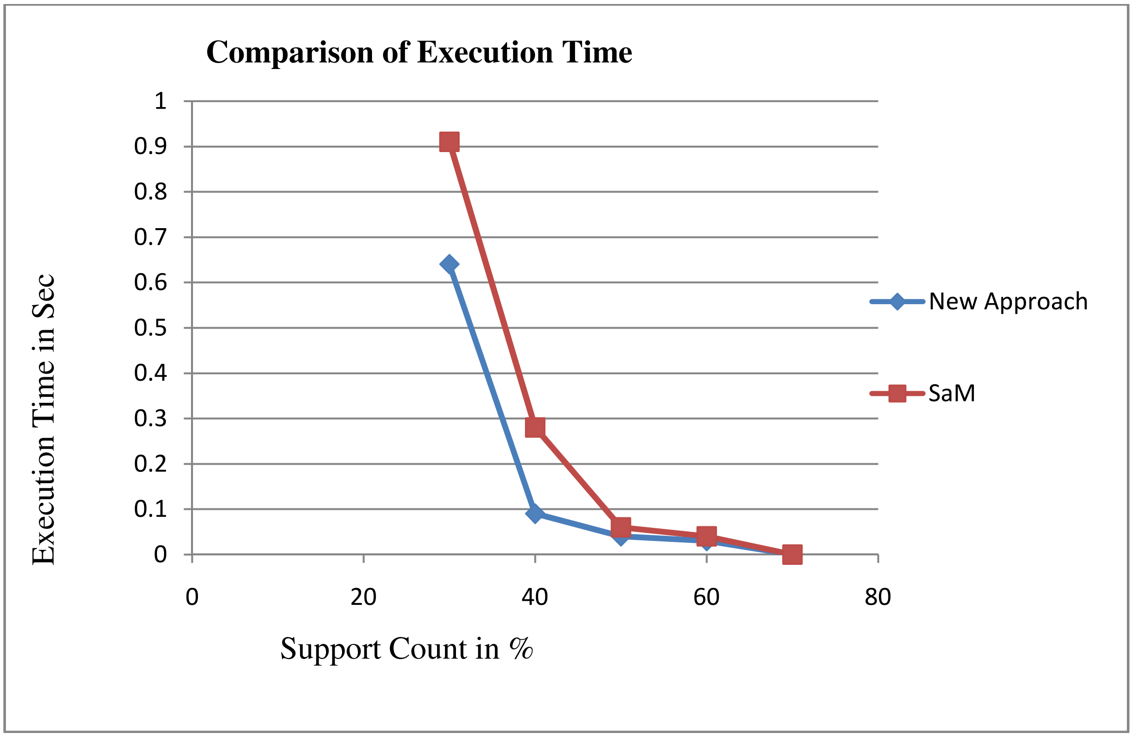


Figure – 3
Execution time for hepatitis dataset

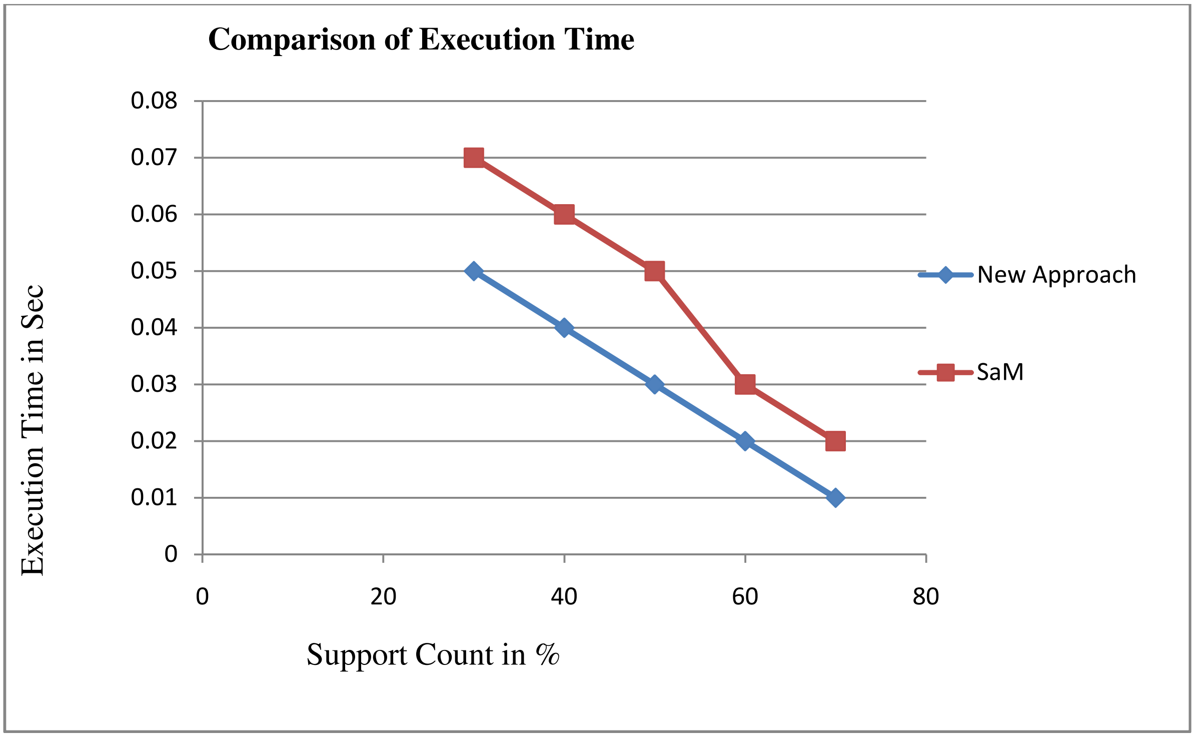


Figure – 4
Execution time for heart dataset

Conclusion

A study of few algorithms is done which made a significant contribution to the search of improving the efficiency of frequent itemset mining algorithm. By analytical study of the classical frequent itemset mining algorithms like Apriori, DHP, partitioning, sampling, DIC, Eclat, FP-growth, H-mine and find out the strength and weaknesses of these algorithms. A new frequent itemsets tree (FI-tree) structure developed, which is used for storing frequent itemsets and their Tid sets. A distinct feature of this method is that it has runs fast in different data characteristics. Our study shows that a new approach has high performance in various kinds of data, outperforms the previously developed algorithms in different settings, and is highly scalable in mining different databases.

References

1. Agrawal R., Imielienski T. and A. Swami, Mining Association Rules between Sets of Items in Large Databases, *Proc. Conf. on Management of Data*, 207–216 **(1993)**

2. Raorane A.A., Kulkarni R.V. and Jitkar B.D., Association Rule – Extracting Knowledge Using Market Basket Analysis, *Res. J. Recent Sci.,* **1**(**2**), 19-27 **(2012)**

3. Shrivastava Neeraj and Lodhi Singh Swati, Overview of Non-redundant Association Rule Mining, *Res. J. Recent Sci.,* **1**(**2**), 108-112 **(2012)**

4. Pramod S. and Vyas O.P., Survey on Frequent Item set Mining Algorithms, *In Proc. International Journal of Computer Applications* (0975 - 8887), **1(15),** 86–91 **(2010)**

5. Agrawal R. and Srikant R., Fast algorithms for mining association rules, *In Proc. Int'l Conf. Very Large Data Bases (VLDB),* 487–499 **(1994)**

6. Park J.S., Chen M.S. and Yu P.S., An effective hash-based algorithm for mining association rules, *In Proc. ACM-SIGMOD Int'l Conf. Management of Data (SIGMOD),* 175–186 **(1995)**

7. Brin S., Motwani R, Ullman J.D. and Tsur S., Dynamic itemset counting and implication rules for market basket analysis, *In Proc. ACM-SIGMOD Int'l Conf. Management of Data (SIGMOD),* 255–264 **(1997)**

8. Savasere A., Omiecinski E. and Navathe S., An efficient algorithm for mining association rules in large databases, *In Proc. Int'l Conf. Very Large Data Bases (VLDB),* 432–443 **(1995)**

9. Toivonen C.H., Sampling large databases for association rules, *In Proc. Int'l Conf. Very Large Data Bases (VLDB),* 134–145 **(1996)**

10. Borgelt C., Efficient Implementations of Apriori and Eclat, *In Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations* **(2003)**

11. Han J., Pei H. and Yin Y., Mining Frequent Patterns without Candidate Generation, *In Proc. Conf. on the Management of Data* **(2000)**

12. Pei J., Han J., Lu H., Nishio S., Tang S. and Yang D., H-mine: Hyper-structure mining of frequent patterns in large databases, *In Proc. Int'l Conf. Data Mining* **(2001)**

13. Borgelt C., SaM: Simple Algorithms for Frequent Item Set Mining, *IFSA/EUSFLAT 2009 conference* **(2009)**

14. Blake C.L. and Merz C.J., UCI Repository of Machine Learning Databases, *Dept. of Information and Computer Science, University of California at Irvine, CA, USA* **(1998)**