



## A Study of Frequent Pattern Mining Methods

Patel Harshit and Jayesh Chaudhary

Sarvajani College of Engineering and Technology, Gujarat, INDIA

Available online at: [www.isca.in](http://www.isca.in), [www.isca.me](http://www.isca.me)

Received 21<sup>st</sup> March 2014, revised 3<sup>rd</sup> April 2014, accepted 19<sup>th</sup> April 2014

### Abstract

*Finding of association rules is a crucial problem in data mining. Two sub-problems of mining association rules. First find out frequent itemsets from dataset and then develop association rules based on frequent itemsets. The important factor is time required for finding frequent itemsets. All the previous algorithms are not efficient and scalable for mining frequent itemsets in transaction datasets. In this paper, we provide an unifying feature for generating frequent itemset algorithms. The performance analysis with Wine, Hepatitis, Heart datasets. The algorithms analysis using different minimum support, number of rows and columns.*

**Keywords:** Data mining, wine, hepatitis, heart.

### Introduction

The aim for finding association rules came from analyze of super market dataset, to find out customer behaviour based on purchased products. Finding of association rules is a crucial problem in data mining. Two sub-problems of mining association rules. First find out frequent itemsets from dataset and then develop association rules based on frequent itemsets.

Many data mining tasks that try to find interesting knowledge from databsets. Apriori algorithm is useful for market basket analysis where many number of transactions but small frequent items<sup>1</sup>. To analyze the huge amount of data thereby exploiting the consumer behavior and make the correct decision leading to competitive edge over rivals<sup>2</sup>. Also Sequential association rule mining is one of the possible methods to analysis of data used by frequent itemsets<sup>3</sup>.

In this paper, we conclude of frequent itemset mining algorithms. The performance analysis with Wine, Hepatitis, Heart datasets. The algorithms analysis using different minimum support, number of rows and columns<sup>4</sup>.

**Problem Description:** More number of frequent itemsets generated in mining of frequent itemsets. If support value is lower, more number of itemsets generated. Once generated more number of itemsets, they all are not interested to end user or analyst. If pruning uninterested itemsets is a time consuming process. The main aim is to optimize the process of discovering itemsets which should be efficient, scalable and can detect only interesting itemsets.

### Methodology

**Frequent Item Graph (FIG) Algorithm:** Find all the frequent itemsets quickly using FIG algorithm<sup>5</sup>. It finds all the frequent itemsets with scanning of one dataset.

**Steps:** i. A full I/O scan of the dataset. The first scan of the dataset discovers the frequent 2-itemsets. ii. A full scan of frequent 2-itemsets. To creating the graphical structure using frequent 2-itemsets.

**Advantages:** i. Does not use candidates. ii. One time scan the dataset.

**Disadvantage:** To creating the graphical structure using full scan of frequent 2-itemsets in second step.

**Frequent Itemsets Algorithm for Similar Transactions (FIASST):** Mining frequent itemsets based on similar transactions after deleting infrequent 1-itemsets using FIASST algorithm<sup>6</sup>. The FIASST algorithm is a pattern growth approach without candidate generation.

**Steps:** i. By single scan of the dataset and aggregated the transactions that have similar itemsets for creating the ItemTable and the BitTable. ii. By FIASST algorithm for finding all frequent itemsets.

**AFPT Algorithm:** The hybrid version of Apriori and FP-growth algorithms is AFPT algorithm<sup>7</sup>.

**Steps:** i. To creating the FP-tree. ii. Using the apriori algorithm for mining FP-tree. To add NTable with two fields named Item-name and Item-support.

**MFIPA Algorithm:** Mining frequent itemsets based on projection array using MFIPA algorithm<sup>8</sup>. The PArray is created using data horizontally and vertically. It stores all the frequent 1-itemsets and those items that co-occurrence with signal frequent item. To avoids the creation of numbers of candidate itemsets are large and redundant detection using PArray.

## Results and Discussion

**Performance Analysis:** The details of Wine, Hepatitis and Heart datasets selected for the experiment from the UCI repository of machine learning databases (table 1)<sup>9</sup>. A detailed study to assess the performance of frequent pattern mining methods. The analysis in the experiments is the total execution time taken and the number of patterns generated using Wine, Hepatitis and Heart datasets. For this comparison also same data sets were selected as for the above experiment with 30% to 70% of minimum support threshold.

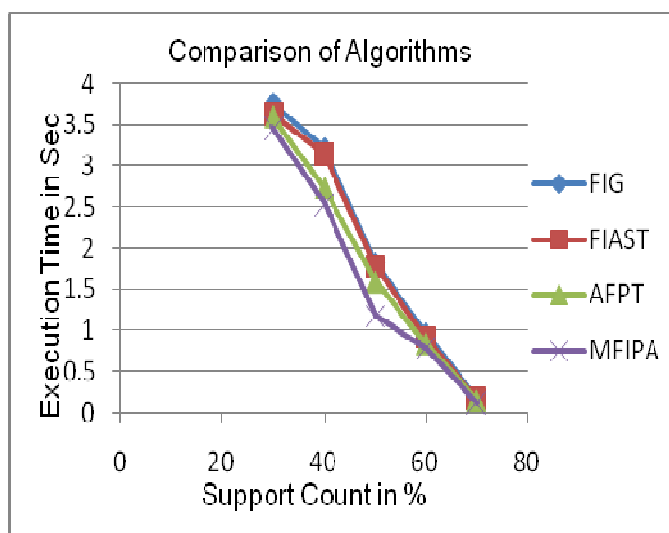
**Table-1**  
**Details of Dataset**

Files	Number of Records	Number of Columns
Wine.data.txt	178	14
Hepatitis.data.txt	155	19
Heart.data.txt	303	75

The execution time for all the algorithms with different support threshold for Wine data set (table 2). The total execution time for the MFIPA and AFPT algorithms small decreases with the increase in support threshold for Wine dataset. The MFIPA takes less time compared to AFPT (figure-1).

**Table-2**  
**Total execution time using Wine dataset**

Support	Total Execution Time in Seconds			
	FIG	FIAS	AFPT	MFIPA
30	3.76	3.63	3.60	3.46
40	3.21	3.14	2.73	2.54
50	1.82	1.77	1.59	1.18
60	0.96	0.91	0.84	0.79
70	0.17	0.16	0.15	0.12

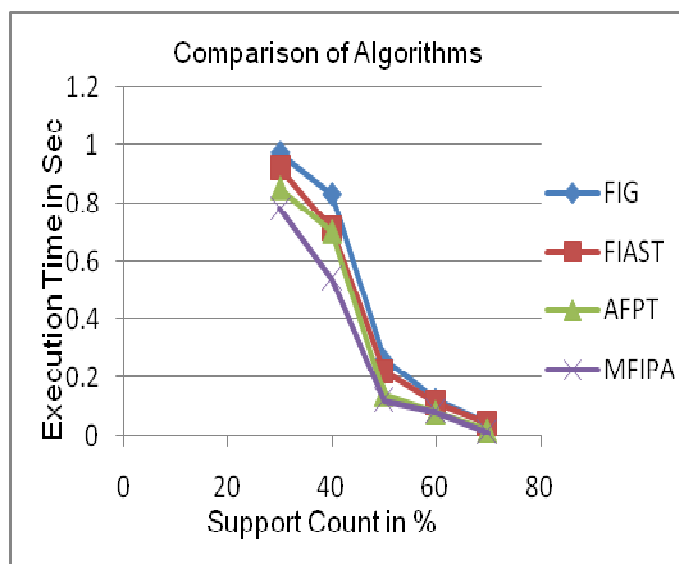


**Figure-1**  
**Execution Time for Wine data set**

The execution time for all the algorithms with different support threshold for Hepatitis data set (table 3). The total execution time for the MFIPA and FIAS algorithms small decreases with the increase in support threshold for Hepatitis dataset. The MFIPA takes less time compared to FIAS (figure-2).

**Table-3**  
**Total execution time using Hepatitis dataset**

Support	Total Execution Time in Seconds			
	FIG	FIAS	AFPT	MFIPA
30	0.97	0.92	0.85	0.78
40	0.83	0.71	0.70	0.54
50	0.26	0.22	0.14	0.12
60	0.12	0.11	0.08	0.08
70	0.04	0.04	0.02	0.01

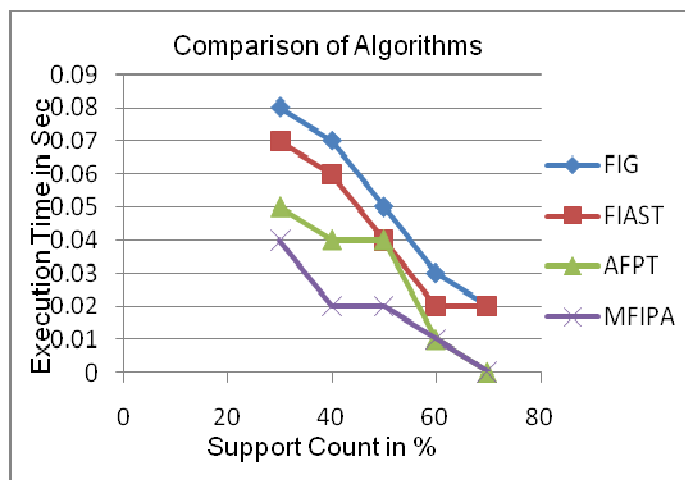


**Figure-2**  
**Execution Time for Hepatitis data set**

The execution time for all the algorithms with different support threshold for heart data set (table 4). The total execution time for the AFPT and FIG algorithms small decreases with the increase in support threshold for Heart dataset. The AFPT takes less time compared to FIG (figure-3).

**Table-4**  
**Total execution time using Heart dataset**

Support	Total Execution Time in Seconds			
	FIG	FIAS	AFPT	MFIPA
30	0.08	0.07	0.05	0.04
40	0.07	0.06	0.04	0.02
50	0.05	0.04	0.04	0.02
60	0.03	0.02	0.01	0.01
70	0.02	0.02	0.00	0.00



**Figure-3**  
**Total execution time using Heart dataset**

Overall the execution time of the FIAST, AFPT and MFIPA algorithms are nearby but it can also be analyzed that the execution time of MFIPA is comparatively less than FIG for higher support threshold. The time of execution is decreased with the increase support threshold.

## Conclusion

We provide a brief overview of FIG, FIAST, AFPT and MFIPA algorithms for generating frequent itemsets. The performance analysis with standard datasets Wine, Hepatitis, Heart etc. are used. By comparing them to classical frequent itemset mining algorithms strength and weaknesses of these algorithms were analyzed.

A comparison of FIG, FIAST, AFPT and MFIPA algorithms. The execution time of the FIAST, AFPT and MFIPA algorithms are nearby but it can also be analyzed that the execution time of MFIPA is comparatively less than FIG for higher support threshold.

## References

1. Agrawal R. and Srikant R., Fast algorithms for mining association rules, *In Proc. Int'l Conf. Very Large Data Bases (VLDB)*, 487–499 (1994)
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. R. Agrawal, T. Imieliński and A. Swami, Mining Association Rules between Sets of Items in Large Databases, *Proc. Conf. on Management of Data*, 207–216 (1993)
5. Kumar A.V.S. and Wahidabanu R.S.D., A Frequent Item Graph Approach for Discovering Frequent Itemsets, *In Proc. Conf. Advanced Computer Theory and Engineering*, 952-956 (2008)
6. Duemong F., Preechaveerakul L. and Vanichayobon S., FIAST: A Novel Algorithm for Mining Frequent Itemsets, *In Proc. Int'l Conf. Future Computer and Communication*, 140-144 (2009)
7. Qihua Lan, Defu Zhang and Bo Wu, A New Algorithm for Frequent Itemsets Mining Based on Apriori and FP-Tree, *Proc. Intelligent Systems*, 360–364 (2009)
8. Hai-Tao He, Hai-Yan Cao, Rui-Xia Yao, Jia-Dong Ren, Chang-Zhen Hu, Mining frequent itemsets based on projection array, *Machine Learning and Cybernetics (ICMLC)*, 454-459 (2010)
9. C.L. Blake and C.J. Merz., UCI Repository of Machine Learning Databases, *Dept. of Information and Computer Science, University of California at Irvine, CA, USA* (1998)