AN ANALYSIS OF FREQUENT PATTERN MINING METHODS

Patel Tushar S.
Assistant Professor, Department Of Information Technology, SPBPEC, Mehsana, Gujarat (India)

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, Itemset, Heart, Hepatitis, Wine

I. 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 [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]. Also Sequential association rule mining is one of the possible methods to analysis of data used by frequent itemsets. 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.

II. 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.

III. METHODOLOGY

3.1 Frequent Item Graph (Fig) Algorithm:

Find all the frequent itemsets quickly using FIG algorithm [3]. It finds all the frequent itemsets with scaning of one dataset. The steps are (1) A full I/O scan of the dataset. The first scan of the dataset discovers the frequent 2-itemsets and (2) A full scan of frequent 2-itemsets. To creating the graphical structure using frequent 2-itemsets. The advantages are (1) Does not use candidates and (2) One time scan the dataset. The disadvantage is to creating the graphical structure using full scan of frequent 2-itemsets in second step.

3.2 Frequent Itemsets Algorithm For Similar Transactions (FIAST):

Mining frequent itemsets based on similar transactions after deleting infrequent 1-itemsets using FIAST algorithm [4]. The FIAST algorithm is a pattern growth approach without candidate generation. The steps are (1) By single scan of the dataset and aggregated the transactions that have similar itemsets for creating the ItemTable and the BitTable and (2) By FIAST algorithm for finding all frequent itemsets.

3.3 AFPT Algorithm:

The hybrid version of Apriori and FP-growth algorithms is AFPT algorithm [5]. The steps are (1) To creating the FP-tree and (2) Using the apriori algorithm for mining FP-tree. To add NTable with two fields named Item-name and Item-support.

3.4 MFIPA Algorithm:

Mining frequent itemsets based on projection array using MFIPA algorithm [6]. The PArray is created using data horizontally and vertically. It stores all the frequent 1-itemsets and those items that co-occurence with signal frequent item. To avoids the creation of numbers of candidate itemsets are large and redundant detection using PArray.

VI. PERFORMANCE ANALYSIS

From table 1, the details of Wine, Hepatitis and Heart datasets selected for the experiment from the UCI repository of machine learning databases [7]. 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

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

Table 2: Total Execution Time Using Wine Dataset

Support	Total Execution Time in Seconds			
	FIG	FIAST	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

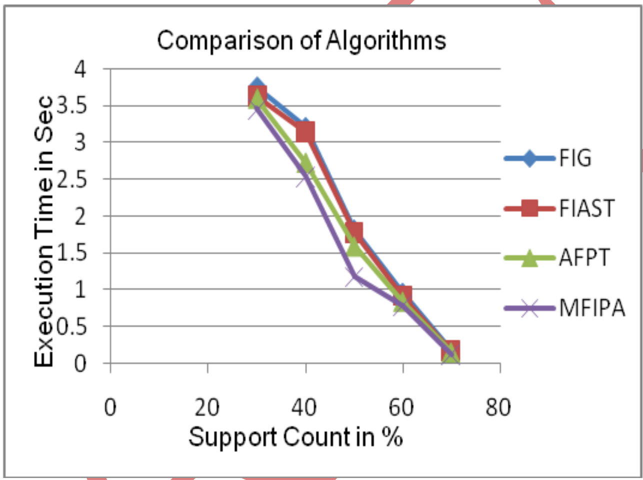

Fig. 1: Execution Time For Wine Data Set

From table 3, the execution time for all the algorithms with different support threshold for Hepatitis data set. The total execution time for the MFIPA and FIAST algorithms small decreases with the increase in support threshold for Hepatitis dataset. From fig. 2, the MFIPA takes less time compared to FIAST.

Table 3: Total Execution Time Using Hepatitis Dataset

Support	Total Execution Time in Seconds			
	FIG	FIAST	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

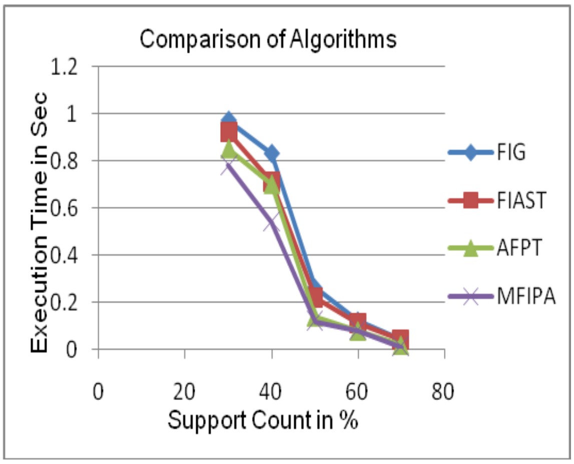

Fig. 2: Execution Time For Hepatitis Data Set

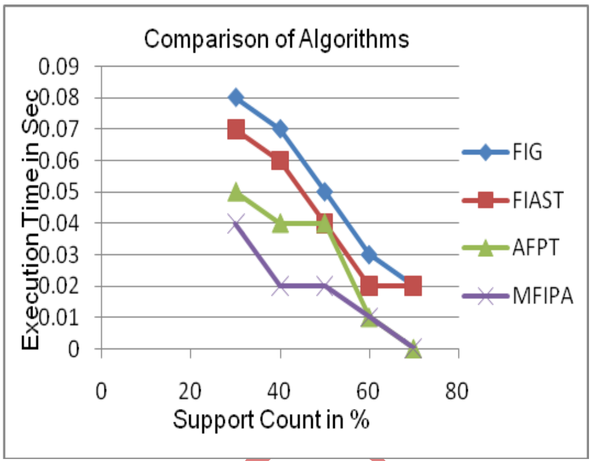

Fig. 3: Execution Time For Heart Dataset

From table 4, the execution time for all the algorithms with different support threshold for Heart data set. The total execution time for the AFPT and FIG algorithms small decreases with the increase in support threshold for Heart dataset. From fig. 3, the AFPT takes less time compared to FIG.

Table 4: Total Execution Time Using Heart Dataset

Support	Total Execution Time in Seconds			
	FIG	FIAST	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

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.

V. CONCLUSION

We provide a brief overview of FIG, FIAST, AFPT and MFIPA algorithms for generating frequent itemsets. The performance analyses 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), 1994, 487–499.

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

[3] Kumar, A.V.S., Wahidabanu, R.S.D., A Frequent Item Graph Approach for Discovering Frequent Itemsets, In Proc. Conf. Advanced Computer Theory and Engineerin, 2008, 952-956.

[4] Duemong, F., Preechaveerakul, L., Vanichayobon, S., FIAST: A Novel Algorithm for Mining Frequent Itemsets, In Proc. Int'l Conf. Future Computer and Communication, 2009, 140-144.

[5] Qihua Lan, Defu Zhang, Bo Wu, A New Algorithm for Frequent Itemsets Mining Based on Apriori and FP-Tree, Proc. Intelligent Systems, 2009, 360–364.

[6] 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), 2010, 454-459.

[7] 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.