ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Comparative Study of Various Frequent Pattern Mining Algorithms

Amit Mittal[1], Ashutosh Nagar[2], Kartik Gupta[3], Rishi Nahar[4]

Assistant Professor, Computer Engineering Dept., Institute of Engineering and Technology (DAVV), Indore, India[1]

Scholar, Computer Engineering Department, Institute of Engineering and Technology (DAVV), Indore, India[2, 3, 4]

Abstract: Frequent pattern mining has become an important data mining task and has been a focused theme in data mining research. Frequent pattern mining aims to find frequently occurring subsets in sequence of sets. The frequent pattern mining appears as a sub problem in many other data mining fields such as association rules discovery, classification, clustering, web mining, market analysis etc. Different frameworks have been defined for frequent pattern mining. The most common one is the support based framework, in which item sets with frequency above a given threshold are found. This paper presents review of different frequent mining algorithms including Apriori, FP-growth and DIC. A brief description of each technique has been provided. In the last, different frequent pattern mining techniques are compared based on various parameters of importance.

Keywords: Data mining; Frequent patterns; Frequent pattern mining (FPM); support; itemset.

I. INTRODUCTION

The algorithmic aspects of Frequent Pattern Mining (FPM) have been explored very widely. The most common one is the support-based framework, in which itemsets with frequency above a given threshold are found. Frequent pattern mining is a first step in association rule mining. The analysis of finding frequent pattern in a database was originally proposed in context of market basket data in order to find the frequent groups of items that are bought together, but now it has also been applied in the context of data mining, web log mining, sequential pattern mining and software bug analysis.

1.1 Support

A transaction T supports an item set I if I is contained in transaction T. Support for an item set I is defined as the ratio of the number of transactions that contain I to the total number of transactions[1].

Let's say if the number of transactions that contain itemset I are M and the total number of transactions are N than the support S can be calculated as S=M / N.

1.2.Support Based Framework

In the support-based framework, in which itemsets with frequency above a given threshold are found. The FPM problem is concerned with finding relationships between different items in a database containing customer transactions

1.3.Frequent Patterns (Frequent Itemset)

If support value of an item set I in the transactional database is greater than the specified threshold value of support than the item set is frequent [1].

Frequent Pattern mining aims to solve the problem of finding relationship among items in a database. The problem can be stated as:

"Given a database D with transactions T1 . . . TN, determine all patterns P that are present in at least a fraction s of the transactions."[7]

The fraction's' here is referred to the minimum support. It can be expressed as an absolute number or as a fraction.

A transaction t is said to contain an item set X if and only if all items within X are also contained in T. Each transaction also contains a unique identifier called Transaction Identification (TID). An item set is considered as frequent or large, if the itemset has a support that is greater or equal to the user specified minimum support.

The number of possible combinations of itemsets increases exponentially with I and the average transaction length. Therefore it is not convenient to determine the support of all possible item sets. When counting the supports of itemsets, there are two strategies. The first strategy is to count the occurrences directly, whenever an item set is contained in a transaction, the occurrence of the item set is increased. The second strategy is to count the occurrences indirectly by intersecting TID set of each component of the item set. The TID set of a component X, where X can be either item or item set, is denoted as X.TID. The support of an item set S = X ∪ Y is obtained by intersecting X.TID ∩Y.TID = S.TID and the support of S equals S.TID. [7]

II. VARIOUS FREQUENT PATTERN MINING TECHNIQUE

2.1.Apriori Algorithms

One of the first algorithms to evolve for frequent itemset mining was Apriori. It was given by R Aggarwal and R Srikant in 1994[1].It works on horizontal layout based database. This algorithm employs an iterative approach known as level wise search. It uses important property called Apriori property is used to reduce the search [2].

DOI 10.17148/IJARCCE.2015.44127

All the non-empty subsets of frequent item sets must also be frequent this property belongs to a special category of properties called Anti-monotone. If a set can't pass a certain test than all of its supersets will fail the same test this property is called anti-monotone.

It generates candidate item sets from scanning database and generates frequent item set by removing all infrequent item set.

Apriori follows two steps approach:
In the first step, it joins two itemsets which contain k-1 common items in kth pass. The first pass starts from the single item, the resulting set is called the candidate set Ck.

In the second step, the algorithm counts the occurrence of each candidate set and prune all infrequent itemsets. The algorithm ends when no further extension found

2.2.FP- growth Algorithm
Frequent pattern growth also labelled as FP-growth is a tree based algorithm to mine frequent patterns in database the idea was given by han et. al. 2000[3] .In FP-growth algorithm database is stored in the form of compact data structure called FP-Tree. It uses divide and conquer method [4]. In it no candidate frequent itemset is needed rather frequent patterns are mined from FP tree. In the first step a list of frequent itemset is generated and sorted in their decreasing support order. This list is represented by a structure called node. Each node in the FP tree, other than the root node, will contain the item name, support count, and a pointer to link to a node in the tree that has the same item name [5].

These nodes are used to create the FP tree. Common prefixes can be shared during FP tree construction. The paths from root to leaf nodes are arranged in non-increasing order of their support. Once the FP tree is constructed then frequent patterns are extracted from the FP tree starting from the leaf nodes. Each prefix path subtree is processed recursively to mine frequent itemsets. FP Growth takes least memory because of projected layout and is storage efficient. A variant of FP tree is conditional FP tree that could be built if we consider transactions containing a particular itemset and then removing that itemset from all transactions.

2.3. Dynamic Itemset count Algorithm
It is an extension to Apriori algorithm which is used to reduce number of scans on the dataset. It is based upon the downward disclosure property.

Alternative to Apriori Itemset Generation.
Itemsets are dynamically added and deleted as transactions are read.Relies on the fact that for an itemset to be frequent, all of its subsets must also be frequent, so we only examine those itemsets whose subsets are all frequent. [6]
In this dynamic blocks are formed from the database marked by start points and unlike the previous techniques of Apriori it dynamically changes the sets of candidates during the database scan. Unlike the Apriori it cannot start the next level scan at the end of first level scan, it start the scan by starting label attached to each dynamic partition of candidate sets.

Itemsets are marked in four different ways as they are counted:

Solid box: confirmed frequent itemset-- an itemset for whichwe have finished counting but it exceeds the threshold minimum support.

- **Solid circle** **:** confirmed infrequent itemset-- an itemset for which we have finished counting and it is below minimum support.
- **Dashed box:** suspected frequent itemset--an itemset we are still counting that exceeds minimum support.
- **Dashed circle** **:** suspected infrequent itemset-an itemset we are still counting that is below minimum support.

III. COMPARISON OF VARIOUS FREQUENT PATTERN MINING TECHNIQUES

Comparison of different FPM algorithms has been done, where various algorithms are been compared against three parameters, number of database scans required for the generation of frequent itemset, the candidate generation technique used and how thealgorithm is sensitive to the change in user parameters i.e. support.

Apriori algorithm use efficient technique for pruning the candidate item sets, but it requiresa lot of computational time as well as multiple dataset scans to generate candidate item sets. DIC provides considerable flexibility by having the ability to add and delete counted itemsets on the fly but it also requires a lot of computational time.

FP-growth algorithm require only two database scans in order to generate frequent patterns. This method use a compact tree- structure to represent the entire database. It does not require candidate generation, which helps in reducing the computational time.

IV. EXPERIMENTAL RESULTS

Following are real life datasets which were taken [8], these are:
Dataset Table

Dataset	Number of Transaction	Number Of Items	Avg. no. of Transactions per item
Medicine	48842	97	15
Letrecog	127881	107	14
Nursery	112304	101	10
Retail	240185	107	16

1) Analysis for Medicine dataset

DOI 10.17148/IJARCCE.2015.44127

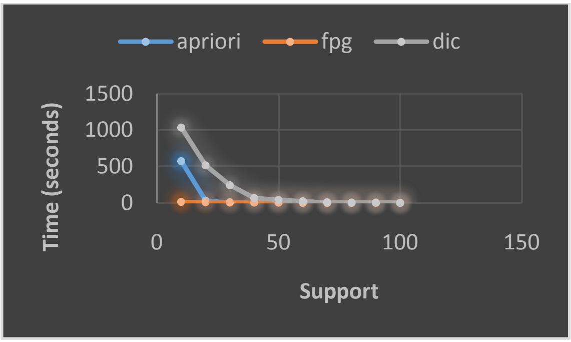

Figure 1: Analysis of algorithms for medicine dataset

Support (%)	Apriori	FP-Growth	DIC
10	572.335	12.68	1032.67
20	31.142	8.12	512.99
30	6.407	4	240.12
40	2.499	3.3	66.49
50	1.76	2.54	37.255
60	1.195	1.4	19.601
70	0.76	0.79	8.322
80	0.595	0.43	4.039
90	0.455	0.3	1.844
100	0.17	0.24	1.156

Table 1: Result of comparison for Medicine dataset

2) Analysis for Letrecog dataset

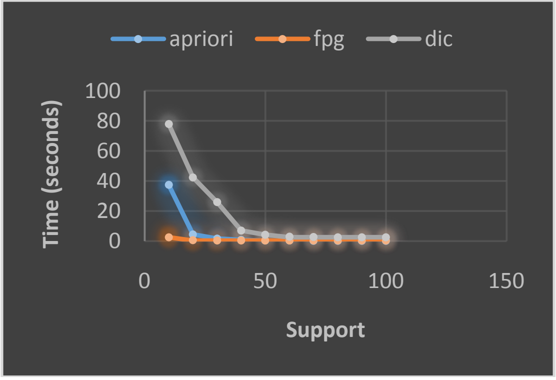

Figure 2: Analysis of algorithms for letrecog dataset

Support (%)	Apriori	FP-Growth	DIC
10	37.549	2.7	77.898
20	4.532	0.77	42.33
30	1.843	0.58	26.099
40	0.875	0.5	7.096
50	0.547	0.47	4.439
60	0.5	0.44	3.063
70	0.328	0.43	2.766
80	0.328	0.43	2.737
90	0.328	0.43	2.68
100	0.328	0.43	2.735

Table 2: Result of comparison for Letrecog dataset

3) Analysis for Nursery dataset

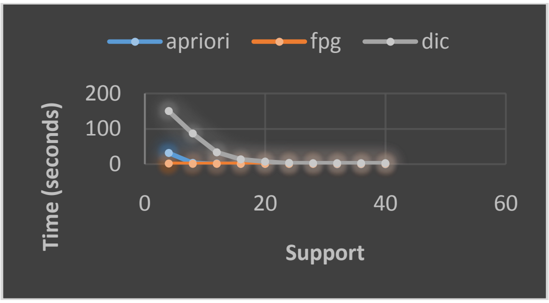

Figure 3: Analysis of algorithms for Nursery dataset

Support (%)	Apriori	FP-Growth	DIC
4	31.722	1.94	150.078
8	3.25	0.84	86.835
12	1.329	0.63	33.778
16	0.89	0.53	13.357
20	0.563	0.41	7.086
24	0.453	0.36	3.565
28	0.406	0.34	2.516
32	0.281	0.34	2.418
36	0.281	0.34	2.345
40	0.281	0.33	2.345

Table 3: Result of comparison for Nursery dataset

4) Analysis for Retail dataset

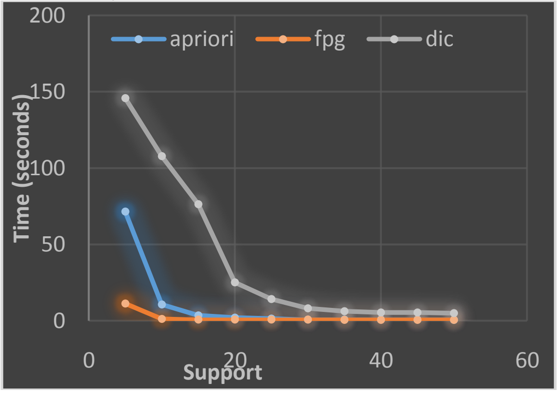

Figure 4: Analysis of algorithms for Retail dataset

Support (%)	Apriori	FP-Growth	DIC
5	71.514	11.31	145.89
10	10.717	1.34	107.78
15	3.644	0.93	76.329
20	2.046	0.85	25.164
25	1.465	0.81	14.212
30	0.931	0.8	8.107
35	0.875	0.67	6.216
40	0.86	0.66	5.552
45	0.855	0.64	5.373
50	0.58	0.6	4.99

Table 4: Result of comparison for Retail dataset

V. CONCLUSION

The Analysis of graphs shows that FP-growth algorithm is the best in all the three algorithms for the experimental datasets [8]. The execution time is decreased when the support threshold gets increased. The performance of FP-Growth, Apriori and DIC algorithm is approximately same for higher value of support. The fastest algorithm for given dataset is FP-growth followed by Apriori. DIC takes more time as compared to other algorithmsfor same datasets.

The scope of the project is very wide because frequent items sets (pattern) are useful for applying various data mining techniques such as classification, clustering, and association rule mining etc. There are different techniques of frequent pattern mining that can be used in different ways to generate frequent itemsets. While working on project, we got the opportunity to understand the broad scope of this field in today's scenario. We have used java platform for implementing algorithms and results so generated could vary with implementing programming language, methodology and machine architecture.

REFERENCES

[1]. Rakesh Agrawal, Ramakrishnan Srikant, Fast Algorithms for Mining

[2]. Association Rules, IBM Almaden Research Center 650 Harry Road, SanJose, CA 95120.

[3]. Julianna KatalinSiposJiawei Han und MichelineKamber. Data Mining – Concepts and Techniques. Chapter 5.2.

[4]. Han, J., Pei, J., and Yin, Y. 2000. Mining frequent patterns without candidate generation. In Proc. 2000 ACMSIGMOD Int. Conf. Management of Data.

[5]. SathishKumar et al. "Efficient Tree Based Distributed Data Mining Algorithms for mining Frequent Patterns" International Journal of Computer Applications (0975 – 8887) Volume 10– No.1, November 2010.

[6]. Rahul Mishra et. al. "Comparative Analysis of Apriori Algorithm and Frequent Pattern Algorithm for Frequent Pattern Mining in Web Log Data." (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (4) , 2012, Pp. 4662 – 4665.

[7]. Tannu Arora ,Rahul Yadav ("Improved Association Mining Algorithm for Large Dataset") International Journal of Computational Engineering and Management VoL 13 July 2011

[8]. Rakesh Agrawal, CC Agrawal, Ramakrishnan Srikant, Analytical Study of Various Frequent Itemset Mining Algorithms http://fimi.ua.ac.be/src/ - Dataset repository

DOI 10.17148/IJARCCE.2015.44127