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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

ALGORITHMS FOR MINING FREQUENT PATTERNS: A COMPARATIVE STUDY

MR.RAJESH K. AHIR[1], MS. MITAL B. AHIR[2]

Assistant Professor, Department of Computer Engineering, G. H. Patel College of Eng & Technology, Anand, Gujarat[1]

M.E. Students, Department of Computer Engineering, Ipcowala Institute of Eng & Technology, Dharmaj, Gujarat[2]

Abstract: Mining frequent patterns are one of the most important research topics in data mining. The function is to mine the transactional data which describes the behaviour of the transaction. In an online business or in an online shopping the customers can purchase items together. Frequent patterns are patterns such as item sets, subsequence or substructures that appear in a data set frequently**.** Many efficient algorithms were developed based on the data structure and the processing scheme. The mining of most efficient algorithms such as Apriori and FP Growth were implemented here. In this paper we propose the efficient algorithms (Apriori and FP Growth) used to mine the frequent patterns. The Apriori algorithm generates candidate set during each pass. It reduces the dataset by discarding the infrequent itemsets that do not meet the minimum threshold from the candidate sets. To avoid the generation of candidate set which is expensive the FP Growth algorithm is used to mine the database.

1 INTRODUCTION

Frequent patterns are patterns such as item sets, subsequence or substructures that appear in a data set frequently. From the transactional database, we can examine the behaviour of the products purchased by the customers. For example a set of items Mobile and Sim card that appear frequently as well as together in a transaction set is a frequent item set. Subsequence means if a customer buys a Mobile he must also buy a Sim card and then head phone etc. From the history of the database these transactions are happening sequentially is called sequential patterns. The Substructure refers to different structural forms such as sub graphs, sub trees which may be used along with item sets or sequences. Many of the algorithms were developed for mining the frequent items.

In this paper we propose the efficient algorithms (Apriori[5][8] and FP Growth[8]) used to mine the frequent patterns. The Apriori algorithm generates candidate set during each pass. It reduces the dataset by discarding the infrequent itemsets that do not meet the minimum threshold from the candidate sets. To avoid the generation of candidate set which is expensive the FP Growth algorithm is used to mine the database. The FP Growth does not generate the candidate set instead it generates an optimized data set that is FP tree from the dataset. The FP tree is mined to construct a conditional database. FP mining processes uses the divide and conquer strategy, so the dataset shrinks and gives us quite small conditional frequent pattern base. From this database the frequent patterns are generated.

2 EXISTING SYSTEM

Traditional system based on the manual calculation and dynamic counting method. Very difficult to find frequent patterns from the large transaction using manual calculation. Dynamic Item set counting method contains very complex procedure. So it is difficult to implement. DIC (Dynamic Item set Counting) algorithm[8][9] which uses more database scan, presents a new approach for finding large item sets. Aim of the DIC algorithm is improving the performance and eliminating repeated database scan. DIC algorithm divides the database into partitions (intervals M) and use a dynamic counting strategy. DIC algorithm determines some stop points for item set counting. Any appropriate points, during the database scan, stopping counting, then starts to count with another item sets.

Disadvantages:

- Time requirement is high.
- User interaction is not efficient.
- Very difficult to process the large transaction.
- It require manual calculation.
- Very complex to implement.

3 PROBLEM DEFINITION

Mining frequent patterns in transactional databases, relational databases, data warehouses, flat files, data streams, advanced database systems include object relational databases and specific application oriented databases such as spatial databases; time series database, text database and multimedia databases have been studied popularly in data mining research. A fundamental problem in data mining is the process of finding frequent patterns in large datasets. Frequent item sets play an essential role in many data mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems.

Data mining system can generate thousands or even millions of patterns or rules. All patterns generated by the system are not meaningful. An interesting pattern represents knowledge. There are several objective measures exist based on the structure of discovered patterns and statistics. Mining Frequent Patterns leads to the discovery of interesting association and correlations within data. If a marketing manager would like to determine which items are purchased together in the same transactions.

IV PROPOSED SYSTEM

We propose the efficient algorithms used to mine the frequent patterns.

1. Apriori Algorithm[5][8].
2. FP Growth[8].

The Apriori algorithm is the most popular association rule algorithm. Apriori uses bottom up search. *FP-Growth* is an algorithm for generating frequent item sets for association rules. This algorithm compresses a large database into a compact, frequent pattern– tree (FP tree) structure. We analysis the time requirement between two algorithms. And make suggestion which one is efficient.

Advantages:

➢ DIC (Dynamic Itemset Counting) is much slower than every other algorithm for the real -dataset. The propose system very fast.

➢ FP growth even has the advantage that the transaction database need not be loaded in ever time.

➢ Very fast compare than existing system.

➢ Candidate key generation is minimum when using FP growth.

IV: IMPLEMENTATION

This implementation contains following modules,

❖ Select Transaction or dataset.

❖ Apriori Implementation.

❖ FP Growth Tree Generation.

❖ Comparison.

a. Select Transaction or dataset:

Through this module we can select the transaction. We will use datasets like Normal DB1, Normal DB2, Plant Cell DB, Snthetic DB, Zoo DB and 1000X8 DB[6]. These values are used to find out the frequent items.

b. Apriori Implementation:

This modules generates the frequent patterns based on apriori algorithm. Execution time of the algorithm this algorithm is noted for the future comparison purpose.

Apriori algorithm works as follows:

• The first step, Apriori algorithm generates Candidate 1 – item sets (C1). Then, item sets count and minimum support value are compared to find the set L1 (frequent item sets).

• The second step, algorithm use L1 to construct the set *C*2 of Candidate 2 – item sets. The process is finished when there are no more candidates.

c. FP Growth Tree Generation:

This module generates the frequent patterns based on FP growth Algorithm. And also it generates the FP Tree to find out the frequent patterns. The FP-growth algorithm is currently one of the fastest approaches to frequent item set mining. It is based on a prefix tree representation of the given database of transactions (called an FP-tree), which can save considerable amounts of memory for storing the transactions.

FP-Growth is an algorithm for generating frequent item sets for association rules. This algorithm compresses a large database into a compact, frequent pattern– tree (FP tree) structure. FP – tree structure stores all necessary information about frequent itemsets in a database. A frequent pattern tree (or FP-tree in short) is defined as

1. The root labeled with “null” and set of items as the children of the root.

2. Each node contains of three fields: item-name (holds the frequent item), count (number of transactions that share that node), and node- link (next node in the FP-tree).

3. Frequent-item header table contains two fields, item-name and head of node link (points to the first node in the FP-tree holding the item).

d. Comparison:

This module generates the bar chart to differentiate the execution time of the two algorithms. Through this module we can easily differentiate the two algorithms.

4 RESULT ANALYSIS

After implementing both the algorithms we have measure the execution time for different databases and generated the bar chart to find an efficient algorithm. The result analysis is shown below.

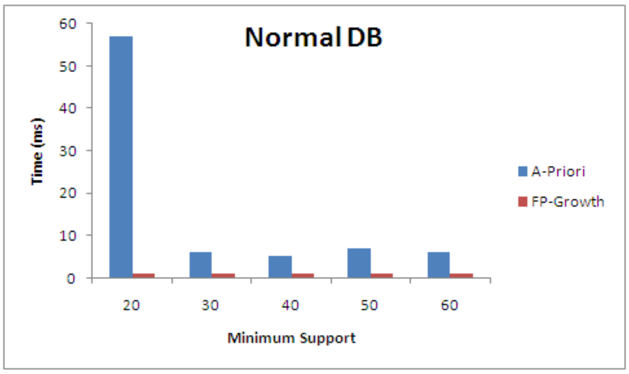

Fig 1 : Normal DB1

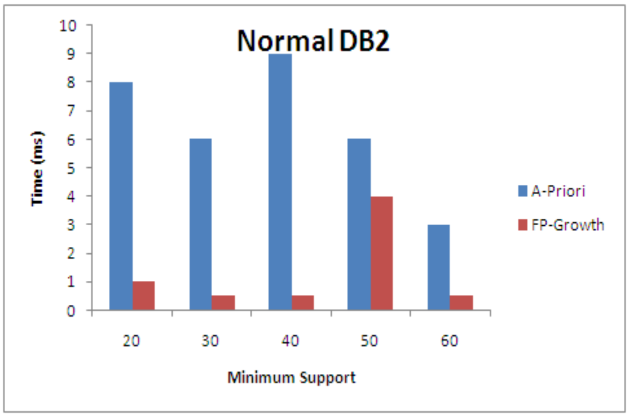

Fig 2 : Normal DB2

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

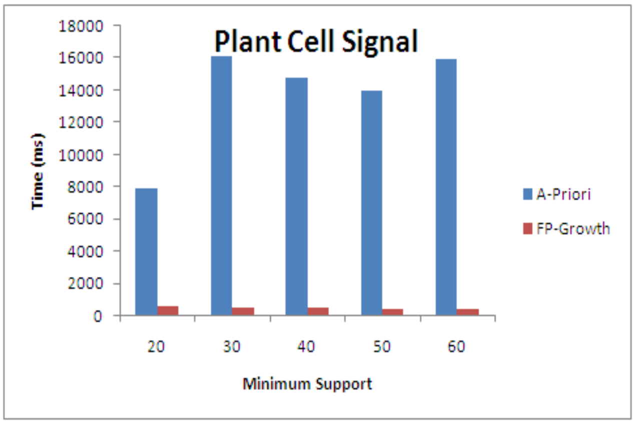

Fig 3 : Plant Cell DB

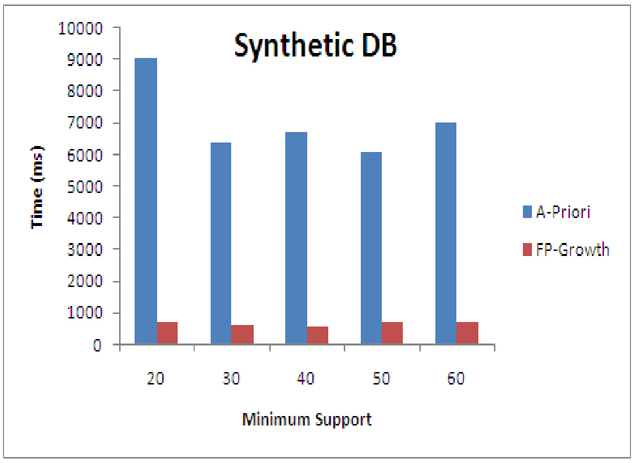

Fig 4 : Synthetic DB

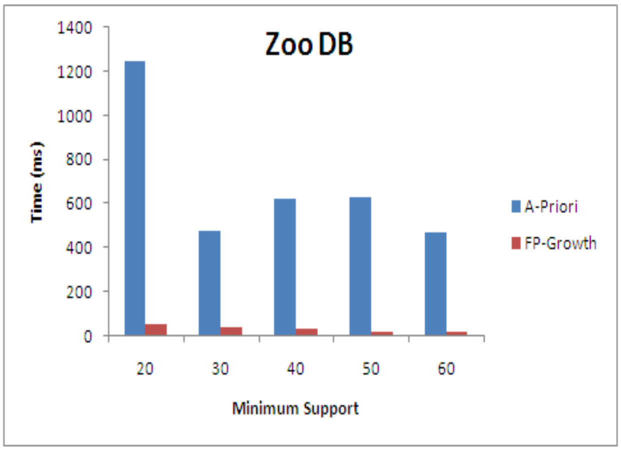

Fig 5 : Zoo DB

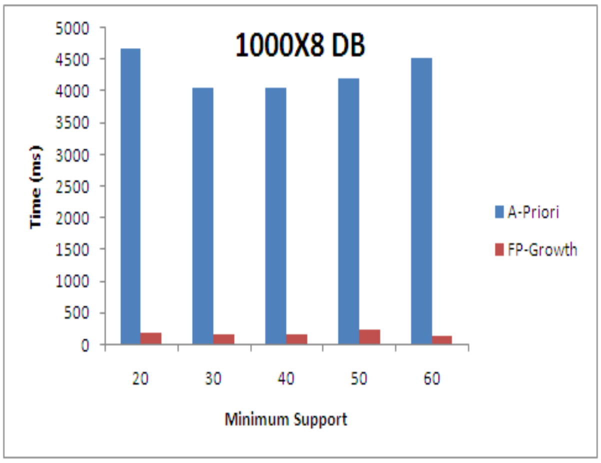

Fig 6 : 1000X8 DB

5 Conclusion

We have implemented both the algorithms using java net beans. We have tested the algorithms on intel core i3 machine with 2GB RAM. A Datasers used here for testing are Normal DB1, Normal DB2, Plant Cell DB, Snthetic DB, Zoo DB and 1000X8 DB. From the results, we can conclude that FP Growth algorithm works better than the Apriori algorithm in terms of Execution time.

REFERENCES

[1]. “Mashroom and Stalog dataset”: http://archive.ics.uci.edu/ml/datasets/
[2]. http://www.java2s.com/Code/Java/J2EE/CatalogJ2ME.htm
[3]http://developers.sun.com/techtopics/mobility/configurations/questions/gcf
[4]. http://en.wikipedia.org/wiki/mining-analysis
[5]. Apriori Ideas: http://www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/itemset_prog1.html
[6]. Dataset file:http://www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/datasets.php
[7]. V. Pudi. “Data Mining: Concepts and Techniques”, Oxford University Press, Jan-2009
[8]. J. Han and Kamber, “Data mining – concepts and techniques”, Elsevier, 2006.
[9]. A.K.Pujari, “Data Mining techniques”, Universities Press, July-2001.
[10]. The Complete Reference Java - Herbert Schildt