International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 10, Issue 6 (June 2014), PP.78-82

Comparison of FP tree and Apriori Algorithm

Prashasti Kanikar, Twinkle Puri, Binita Shah, Ishaan Bazaz, Binita Parekh

Abstract:- Data Mining is known as a rich tool for gathering information and frequent pattern mining algorithm is most widely used approach for association rule mining. To harness this power of mining, the study of performance of frequent pattern algorithm on various data sets has been performed. Using Java as platform implementation of Frequent Pattern Algorithm has been done and analysis is done based on some of the factors like relationship between number of iterations and number of instances between different kinds of data sets.
The drawbacks of the algorithm were studied and a better algorithm was analysed to find out the differences in the performance of the two algorithms namely Frequent Pattern Mining and Partition Algorithm enabling comparative study of those algorithms. Conclusion is supported with graphs at the end of the paper.

Keywords:-Association Rule Mining, Confidence, Data Mining, Data Warehousing, and Knowledge Discovery Process.

I. INTRODUCTION

This review paper presents a detailed study about the Frequent Pattern growth algorithm and mentions the various drawbacks associated with it. A better algorithm with respect to performance has been studied and implemented using the same dataset to give a better understanding of the efficiency of the two algorithms. It provides empirical evidence about the performance with the help of various graphical and tabular data as well as suggests further improvements that can be made in the Partition algorithm for increased performance.

II. WHAT IS DATA MINING

Mining is that fundamental component acting upon a Data Warehouse that helps lay user extract vital information from the huge chunk of data which holds relevance to the kind of information one is looking out for. Mining helps the user narrow down to that specific item in the Data Set that is of consideration for the particular scenario. It is extremely important to implement an efficient mining utility. In simple terms mining is similar to query firing on a normal database table, but the difference lies in the size of the data.
Based on various algorithmic techniques several types of data mining algorithms are implemented and studied constantly to serve the purpose and simultaneously improvise their performance in the near future.
Various Data Mining Algorithms –

- Association Rule Mining
- Clustering
- Anomaly Detection
- Classification
- Regression
- Summarization

Let us look at the detailed explanation of these algorithms with the examples:

1. Association Rule Mining:

Association rule mining is a popular and well researched method for discovering interesting relations between variables in large databases or Data Warehouse. It is intended to identify strong rules discovered in databases using different facts of interest and thus obtain relationships among different items in database.

2. Clustering:

It is defined as the process of grouping similar objects or things into groups or clusters. Clusters in turn can be defined as a subset of objects in a database that have some similar characteristic or property.

3. Anomaly Detection:

Unlike clustering which groups similar objects or creates clusters of objects having similar characteristic, Anomaly detection finds out objects that do not conform to a given pattern. These patterns of objects which do not conform to the normal stated patterns are called Anomaly.

4. Classification:

There is a fine line of difference between classifications a clustering that is it consists of supervised classes in which we know the classes before-hand.

5. Regression:

Regression is a data mining technique which is used to fit an equation to a dataset. It aims to find correct values so as to fit a given equation. The most commonly used Regression technique is Linear Regression.

6. Summarization:

It helps to map data into subsets which also include simple descriptions. They include generalization and characterization.

III. ASSOCIATION RULE MINING (FP GROWTH ALGORITHM)

The FP-Growth Algorithm is an alternative to finding frequent item sets without using candidate generations, thus improving performance. The main reason behind this fact is that it uses a divide-and-conquer strategy. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the item set association information.

The Working of the Algorithm is as follows: First it compresses the input database creating an FP-tree instance to represent frequent items. After this first step it divides the compressed database into a set of conditional databases, each one associated with one frequent pattern. Finally, each such database is mined separately. Using this strategy, the FP-Growth reduces the search costs looking for short patterns recursively and then concatenating them in the long frequent patterns, offering good selectivity.

In large databases, it’s not possible to hold the FP-tree in the main memory. A strategy to cope with this problem is to firstly partition the database into a set of smaller databases (called projected databases), and then construct an FP-tree from each of these smaller databases.

- **Understanding the FP Tree Structure:**

The frequent-pattern tree (FP-tree) is a compact structure that stores quantitative information about frequent patterns in a database.

One root labeled as “null” with a set of item-prefix subtrees as children, and a frequent-item-header table.

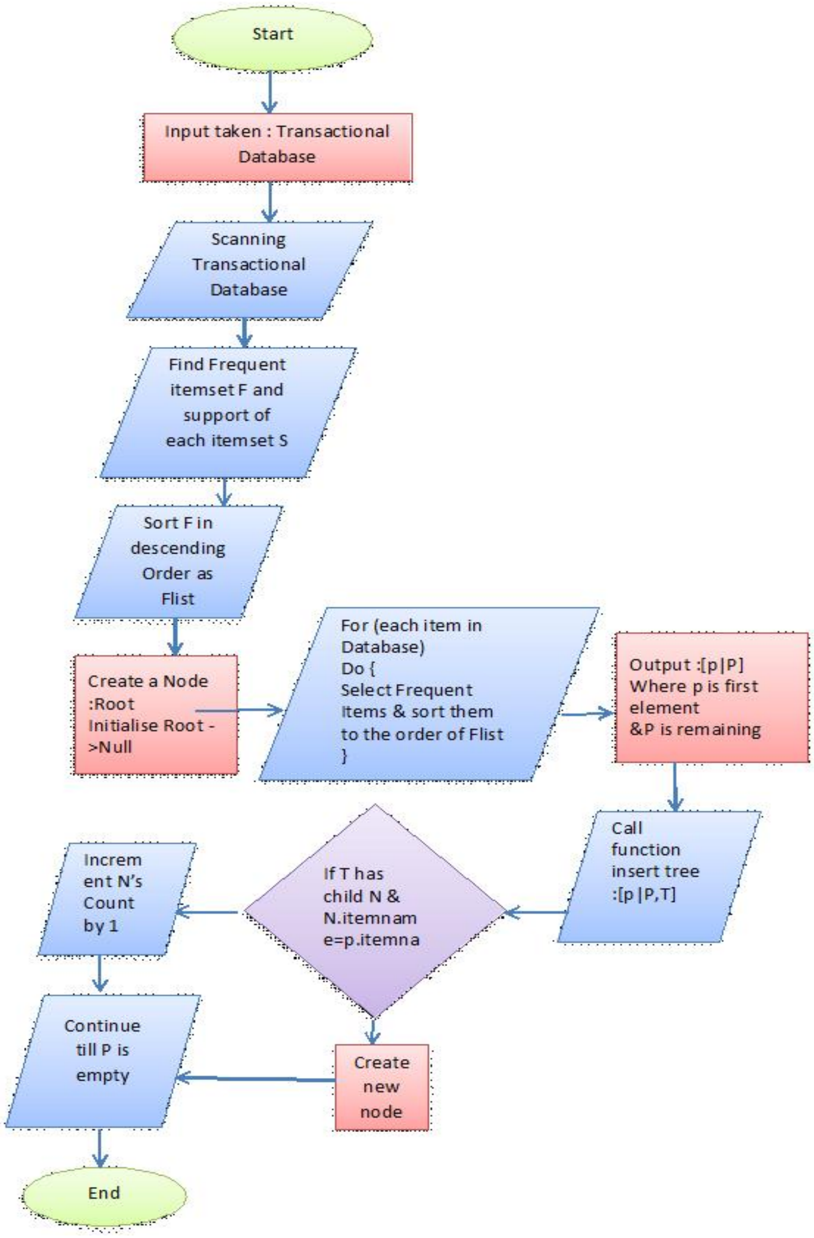
- Each node in the item-prefix subtree consists of three fields:
  - **Item-name**: registers which item is represented by the node;
  - **Count**: the number of transactions represented by the portion of the path reaching the node;
  - **Node-link**: links to the next node in the FP-tree carrying the same item-name, or null if there is none.
- Each entry in the frequent-item-header table consists of two fields:
  - **Item-name:** as the same to the node;
  - **Head of node-link:** a pointer to the first node in the FP-tree carrying the item-name.
- Additionally the frequent-item-header table can have the count support for an item, The Figure below Shows a Representation of the FP-Tree.

The Major differentiation between Apriori and FP Tree Algorithm can be listed as below –

Apriori	FP Tree
Principle : Uses generate and test approach *If an item set is frequent ,then all its subset must be frequent	**Principle** :Allows frequent set generation without candidate set generation
Advantages : For large Transactional Databases	**Advantages :** Compact Data structure is created
Disadvantages : *Generation of Candidate sets is Expensive *Support Counting is expensive	**Disadvantages:** Resulting FP-Tree is not unique for same logical database
Time Scope: Because of Large no. of Scans for determining rules of association, it takes relatively longer to complete the mining process.	**Time Scope:** Comparatively faster than Apriori Algorithm as the scanning process is just carried out once.
Space: The amount of Memory Required to store the data and iterations is large due to larger no. of iterations.	**Space:** The Size of the so formed FP-Tree is smaller than the Original Database, resulting in a much lesser memory requirement.

IV. IMPLEMENTING THE ALGORITHM

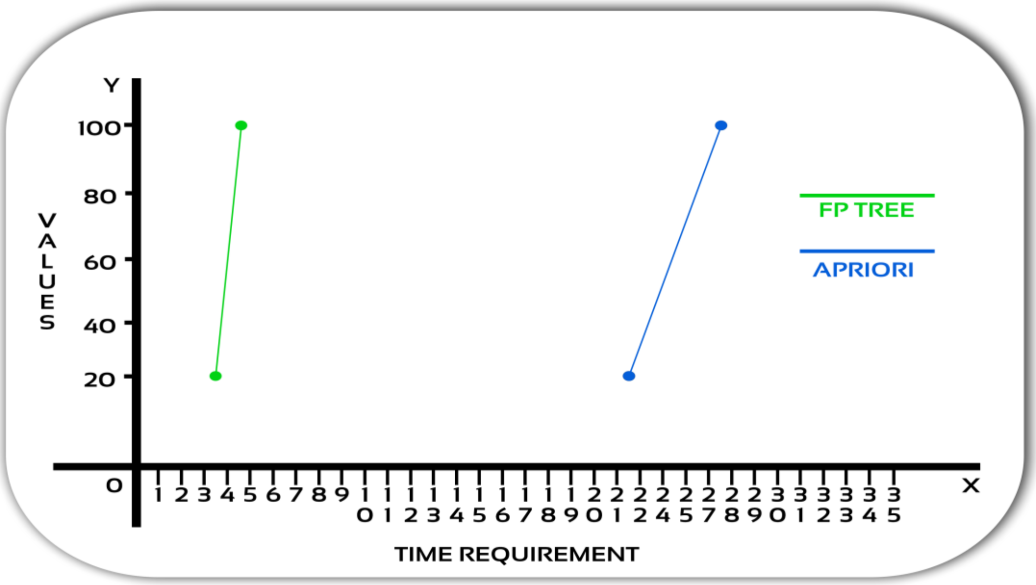
The FP-Tree Algorithm implementation is executed as described below in the flow diagram-


Start
Input taken : Transactional Database
Scanning Transactional Database
Find Frequent itemset F and support of each itemset S
Sort F in descending Order as Flist
Create a Node :Root Initialise Root ->Null
For (each item in Database) Do { Select Frequent Items & sort them to the order of Flist }
Output :[p|P] Where p is first element &P is remaining
Call function insert tree :[p|P,T]
If T has child N & N.itemnam e=p.itemna
Increm ent N's Count by 1
Create new node
Continue till P is empty
End

V. ANALYSIS ON THE RANDOM DATA SET

The Experiment with the same dataset was carried out on both the association rule mining algorithms, here are the results based on time requirement for obtaining the number of association rules as defined before running the algorithm.

	Apriori (Time in Seconds)	FP-Tree (Time in Seconds)
No. of Rules – 20	21.64	3.29
No. of rules – 100	27.59	4.79



Based on the above tabular data and graphical representation, we can decipher with great ease that the FP-Tree algorithm is almost as 8 times faster than the Apriori Algorithm for Association Rule Generation. This directly impacts positively on the Time and Space Complexity Requirement as compared to that of the Apriori Algorithm, proving it to be far more superior and qualitative in terms of resulting output.

VI. CONCLUSIONS

Hence we conclude that for association rule mining of larger datasets FP Tree algorithm Proves to be the better option in comparision to Apriori algorithm because of the advantages of FP Tree over Apriori :

- **Completeness** – FP tree is complete i.e.it does not break any rule in between.
- **Compactness** – FP Tree is compact in a way that size of the Tree structure is smaller than even the original Database used.
- **No. of Scans**: No.of scans performed for finding the frequent items and their support is small i.e only one scan is done for finding the frequent itemsets as compared to Apriori algorithm.
- **Elimination of candidate set generation** : Here there is no candidate set generation.

From the analysis done it can be concluded that the time taken by FP Tree to build association rules is extremely small as compared to Apriori algorithm.

But still these are not feasible in real world application so for that another algorithm which is an improvement for both the algorithms was studied and following advantages of the "Partition Algorithm" was found out over both the classical association rule mining algorithm.

Lower C.P.U time:

In the previous algorithms the time taken by the central processing unit was very large. This includes loading the data, performing various scans on this data and also then generating new data with respect the old data. Also after the final data set has been generated the association rules also require to check the database thus increasing the time further. Moreover this caused a great delay in performing other tasks and thus a newer and better method had to be devised.

This new algorithm has helped to reduce the central processing unit by a factor of 4 which is very significant in performing other tasks and enables better working and usage of the central processing unit.

Lesser memory space requirement:

The previously known algorithms had a very severe drawback of occupying large chunks of memory and thus were very tedious and troublesome to use for a large dataset.

Hence this algorithm had been designed taking into consideration the previous drawback and rather than requiring multiple scans over the database it just requires two scans for generation of the association rule.

Input Output device factor reduced in magnitude:

The earlier known algorithms had, along with huge requirements in database, a much stressed usage of I/O devices which restricted their use for other activities or resulted in large waiting time. With the modern approach to association rule mining the usage of I/O which used to be in greater magnitudes has been lowered to lesser magnitudes thereby enabling access to others.

ACKNOWLEDGMENT

This experiment was successfully accomplished under the sole guidance of our esteemed mentor Prof. Prashasti Kanikar, Our Sincere Thanks and deep gratitude to her for her excellent mentorship, motivating support and thorough aid in understanding the subject and its complexities.

REFERENCES

[1]. A Comparative study of Association rule Mining Algorithm – Review Cornelia Győrödi*, Robert Győrödi*, prof. dr. ing. Stefan Holban

[2]. Image Processing and Image Mining using Decision Trees - KUN-CHE LU AND DON-LIN YANG

[3]. IMAGE MINING TECHNIQUES: A LITERATURE SURVEY - Mahendra S.Makesar

[4]. An Experiential Survey on Image Mining Tools, Techniques and Applications – C. Laxmi Devasena

[5]. http://ijana.in/papers/6.11.pdf

[6]. http://www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/itemset_apriori.html

[7]. http://staffwww.itn.liu.se/~aidvi/courses/06/dm/lectures/lec7.pdf

[8]. http://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=image+mining +algorithms &x=0&y=0

[9]. http://www2.ims.nus.edu.sg/preprints/2005-29.pdf

[10]. http://staffwww.itn.liu.se/~aidvi/courses/06/dm/lectures/lec7.pdf