Comparing the Performance of Frequent Pattern Mining Algorithms

Dr. Kanwal Garg
Assistant Professor, Dept. of Computer Science and Applications, Kurukshetra University, Kurukshetra, Haryana, India

Deepak Kumar
M.Tech Research Scholar, Dept. of Computer Science & Applications, Kurukshetra University, Kurukshetra, Haryana, India

ABSTRACT

Frequent pattern mining is the widely researched field in data mining because of it's importance in many real life applications. Many algorithms are used to mine frequent patterns which gives different performance on different datasets. Apriori, Eclat and FP Growth are the initial basic algorithm used for frequent pattern mining. The premise of this paper is to find major issues/challenges related to algorithms used for frequent pattern mining with respect to transactional database.

General Terms
Algorithms.

Keywords
Data Mining, Frequent Pattern Mining.

1. INTRODUCTION

Data mining has long been an active area of research in databases. The day by day decreasing cost and compactness of storage devices has made it possible to store every transaction of a transactional database [2]. This storage solves two problems first they can access the data any times second this data helps them to find relationship among data items. The problem of finding relationship among different data items was first introduced by agarwal et. al.[1]. The solution to this problem can help to enhance the earnings, optimized storage. In this section the researcher introduces the concept of transactional database, database layout, frequent pattern, frequent itemset and candidate itemset. A database is a systematically arranged collection of data, so that it can be retrieved and manipulated easily at a later time. There are different kinds of database, like active database, cloud database, embedded database and transactional database etc, but in this paper the researcher deals with transactional database only. A transactional database is a database in which there is no auto commit. Most modern relational database are the transactional database [3]. A database layout tells how data is represented. There are two layout which are in common use, horizontal layout and the vertical layout. In horizontal layout there are two columns. First represents the transaction id and second represents the items bought in that transaction. In vertical layout the first column represent the item id and the second shows the transactions id in which the particular item is bought. There is a third layout also known as projected layout. This is not a physical layout. In this layout the system records only the transaction identifier and associated item. It is a divide and conquer mechanism which reduces the size of database recursively by considering only the longest pattern. A frequent pattern is a pattern which occurs in comparatively more transactions. A frequent itemset is an itemset whose support is greater than some user-specified minimum support. The presented paper is organized in five sections: the first section contains the introduction, the second section presents a brief description of the three frequent pattern mining algorithms namely Apriori, Eclat and FP Growth. The third section gives the methodology used. The fourth section presents a comparative analysis of the algorithms used under varying conditions. Fifth section gives the conclusion and in the last references are listed.

2. FREQUENT PATTERN MINING ALGORITHMS

Now the researcher elaborate the various frequent itemset mining algorithms.

2.1 Apriori Algorithm

Apriori is the very first algorithm for mining frequent patterns. It was given by R agarwal and R srikant in 1994 [5].It works on horizontal layout based database. It is based on Boolean association rules which uses generate and test approach. It uses BFS (breadth first search). Apriori uses frequent k itemsets to find a bigger itemset of k+1 items. In Apriori support count for each item is given, the algorithm first scan the database to find out all frequent items based on support. The calculation of frequency of an item is done by counting it's occurrence in all transactions [6]. All infrequent items are dropped.
Apriori property: All subsets of a frequent itemsets which are non empty are also frequent.
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 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 Eclat Algorithm

Eclat is a vertical database layout algorithm used for mining frequent itemsets. It is based on depth first search algorithm. In the first step the data is represented in a bit matrix form. If the item is bought in a particular transaction the bit is set to one else to zero. After that a prefix tree needs to be constructed. To find the first item for the prefix tree the algorithm uses the intersection of the first row with all other rows, and to create the second child the intersection of the second row is taken with the rows following it [4]. In the similar way all other items are found and the prefix tree get constructed. Infrequent rows are discarded from further calculations. To mine frequent itemsets the depth first search algorithm is applied to prefix tree with

backtracking.. Frequent patterns are stored in a bit matrix structure. Eclat is memory efficient because it uses prefix tree. The algorithm has good scalability due to the compact representation.

2.3 Fp Growth Algorithm

Frequent pattern growth also labeled as FP growth is a tree based algorithm to mine frequent patterns in database the idea was given by (han et. al. 2000) [10].It is applicable to projected type database. It uses divide and conquer method [7]. 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 [6]. 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 would be built if we consider transactions containing a particular itemset and then removing that itemset from all transactions. Another variant is parallel fp growth (PFP) that is proposed to parallelize the fp tree on distributed machines [8]. FP Growth is improved using prefix-tree-structure, Grahne and Zhu [9].

3. METHODOLOGY

The above mentioned three algorithms were implemented in java and there performance was compared on synthetic dataset by varying number of attributes and instances. The performance comparing parameter is execution time.

4. COMPARATIVE ANALYSIS

The comparative analysis of the algorithms is shown below by varying various parameter.

Figure 1. Comparison of Apriori, Eclat and FP Growth algorithm on artificial dataset.

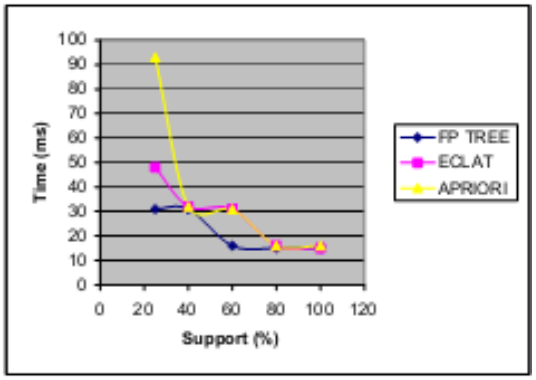


Figure 2.Comparison of Apriori, Eclat and FP Growth algorithm on artificial dataset when the number of transactions are made three times the original.

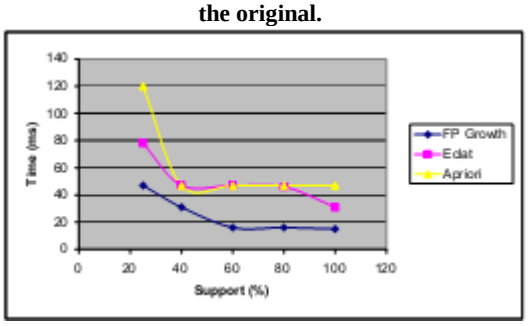


Figure 3. Comparison of apriori, éclat and FP Growth algorithm on artificial dataset when the number of attributes are made three times the original.

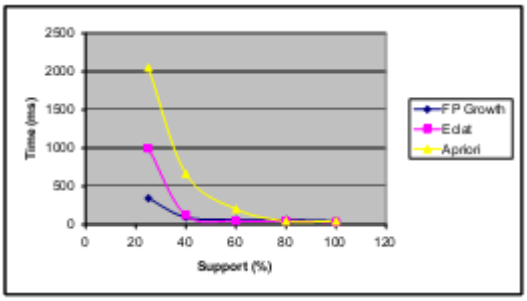


Figure 4. Comparison of Apriori, Eclat and FP Growth algorithm on a dataset with the number of transactions three times the original and number of attributes three times the original.

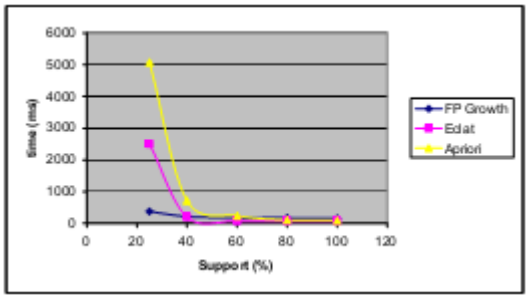


Figure 5. Scaling of the FP Growth with respect to the number of varying attributes and transactions.

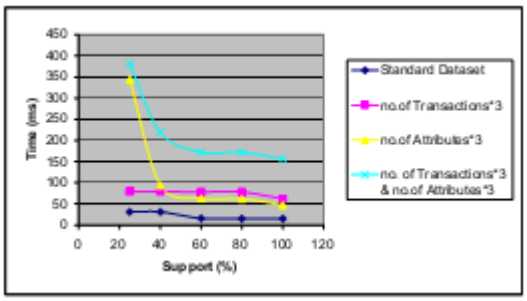


Figure 6. Scaling of Eclat with number of varying attributes and transactions.

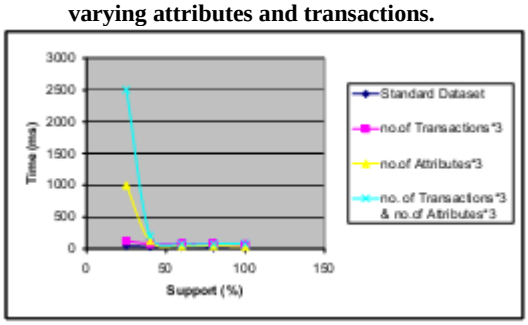
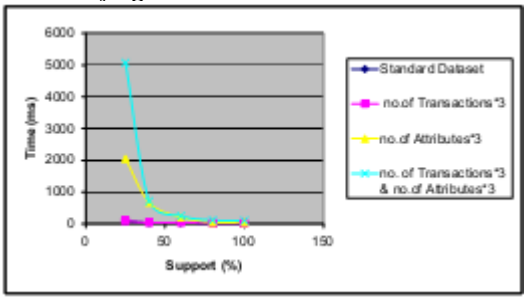


Figure 7. Scaling of the Apriori with number of varying attributes and transactions.



4. RESULT AND DISCUSSION

On standard dataset FP Growth performs the best and Apriori takes the maximum time. When the number of transactions are made three times the original, there is more increment in time for both FP Growth and Eclat than Apriroi. When the number of attributes are made three time the original, FP Growth performs the best and Apriori shows a sharp increase while Eclat lies in the between the two. Also from figure 2 and figure 3 it is clear that increasing the number of attributes affects more any individual algorithm than increasing the number of tuples by same factor. When the number of transactions are made three times and the number of tuples are also increased three times, there is sharp increase in time taken by all algorithms. At this stage FP Growth performs the best and Apriori performs the worst as shown in figure 4. Figure 5 shows the scaling of FP Growth algorithms under different conditions. Figure 6 and figure 7 shows scaling of éclat and apriori under different conditions. From the comparison of figure 5, figure 6 and figure 7 it is clear that FP Growth performs well under all kinds of variations and therefore is the best among three algorithms. Apriori performs the worst among three algorithms and thus shows least scalability, Where as éclat lies in the middle of FP Growth and Apriori.

5. CONCLUSION

Frequent pattern mining is the most important step in association rules which finally helps us in many applications like market basket analysis, clustering, series analysis, games, decision making, object mining, website navigation etc. In this paper the researcher surveyed the pattern mining algorithms namely apriori, Eclat and FP Growth. It is found that apriori uses join and prune method, Ecalt works on vertical datasets and FP Growth constructs the conditional frequent pattern tree which satisfy the minimum support.

The major weakness of Apriori algorithm is producing large number of candidate itemsets and large number of database scans which is equal to maximum length of frequent itemset [5]. It is very much expensive to scan large database[11]. A true reason of apriori failure is it lacks efficient processing method on database [7]. FP Growth is the best among the three algorithms and is thus most scalable. Eclat performs poorer than FP Growth and the Apriori performs the worst. In the future improvements must be taken care to enhance the performance of Apriori and Eclat using a better layout to store the data.

6. REFERENCES

[1] Agarwal, R.C., Agarwal, C.C. and Prasad, V.V.V. (2001) A tree projection algorithm for generation of frequent item sets. Journal of Parallel and Distributed Computing, 61(3), Pp. 350–371.

[2] Bhadoria et. al. Analysis of Frequent Itemset Mining on Variant Datasets published in int.J.comp. Tech.appl., vol(2)5, ISSN:2229-6093, Pp. 1328-1333.

[3] http://en.wikipedia.org/wiki/Database_transaction [on 11th nov 2012].

[4] C.Borgelt. “Efficient Implementations of Apriori and Eclat”. In Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations, CEUR Workshop Proceedings 90, Aachen, Germany 2003.

[5] Goswami D.N et. al. “An Algorithm for Frequent Pattern Mining Based On Apriori ” (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, Pp. 942-947.

[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] 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.

[8] Haoyuan Li,Yi Wang,Dong Zhang, Ming Zhang,Edward Chang 2008."Pfp: parallel fp-growth for query recommendation Proceedings of the 2008 ACM conference on Recommender systems Pp. 107-114.

[9] G. Grahne and J. Zhu , May 2003. "High performance mining of maximal frequent itemsets", In SIAM'03 Workshop on High Performance Data Mining: Pervasive and Data Stream Mining.

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

[11] Deepak Garg et. al. "Comparative Analysis of Various Approaches Used in Frequent Pattern Mining" (IJACSA) International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence.