



Performance Analysis of Association Rule Mining Algorithms

Gagandeep Kaur*

M.Tech, Research Scholar

Department of Computer Science and Engineering
Sri Guru Granth Sahib World University
Fatehgarh Sahib, Punjab, India

Shruti Aggarwal

Assistant Professor

Department of Computer Science and Engineering
Sri Guru Granth Sahib World University
Fatehgarh Sahib, Punjab, India

Abstract— In recent years, Data Mining is an important aspect for generating association rules among the large number of itemsets. Association Rule Mining is the method for discovering interesting relations between variables in large databases. It is considered as one of the important tasks of data mining intended towards decision making. Several association rule mining algorithms have been proposed to generate association rules from the given dataset. The most common algorithms of association rule mining are Apriori and FP-Growth. This paper presents the performance comparison of Apriori and FP-Growth algorithms. The two algorithms are compared based on the execution time and number of scans for different number of instances. The performance study shows that the FP-growth method is efficient and faster than the Apriori algorithm.

Keywords— Data Mining, Association Rule Mining, Support, Confidence, Apriori Algorithm, FP-Growth Algorithm

I. INTRODUCTION

Data Mining is a process of extraction of useful information from huge amount of data. Data mining is often treated as synonym for another popularly used term, Knowledge Discovery in Databases (KDD). The rapid development of information technology in various fields of human life leads to generate large amount of data. The data can be stored in various formats like records, documents, images etc. The data collected from different applications require proper mechanism of extracting knowledge from large repositories for better decision making. Data Mining aims at the discovery of useful information from large collections of data [1]. The association rule mining as an important component of data mining attracts many attentions. Discovering association rules is at the heart of data mining. Association rule mining, which is widely used in medicine, biology, business and so on, is introduced by R. Agrawal et al. in 1993 [2]. From then on, association rules attracted a lot of interest, lots of researchers worked on it.

II. ASSOCIATION RULE MINING

Association Rule Mining is the process of finding interesting correlations, frequent patterns or associations among sets of items in the transaction databases, relational databases or other information repositories. An association rule is an expression in the form of $X \Rightarrow Y$, where X and Y are set of items called itemsets and intersection of X and Y is null [3]. The portion of the rule to the left of the implication (\Rightarrow) is known as the antecedent (X), whereas the right side of the implication is known as the consequent (Y). A rule may contain more than one item in antecedent and consequent part.

Association rule mining tends to produce a large number of rules. The goal is to find the rules that are useful to users. There are two important basic measures for association rules: Support and Confidence. Usually thresholds of support and confidence are predefined by users to drop those rules that are not so interesting or useful. The two thresholds are called minimum support and minimum confidence respectively.

1. **Support** is the percent of the transactions that contain $X \cup Y$ (i.e. both X and Y) to the total number of transactions in database. Suppose the support of an item is 0.1%, it means only 0.1 percent of the transaction contain that item [4].
2. **Confidence** is the percent of the transactions that contain $X \cup Y$ to the total number of transactions that contain X. Suppose the confidence of the association rule $X \Rightarrow Y$ is 70%, it means that 70% of the transactions that contain X also contain Y [4].

A large number of association rule mining algorithms have been developed with different mining efficiencies. Some of the well-known algorithms are Apriori and FP-Growth.

III. APRIORI ALGORITHM

The first algorithm for mining all frequent itemsets and association rules was the AIS algorithm. Shortly after that the algorithm was improved and renamed Apriori. The Apriori algorithm is a classic algorithm for mining all frequent itemsets and association rules. Apriori is designed to operate on databases containing transactions. Each transaction contains set of items called itemset.

Apriori uses level-wise search where k-itemsets (an itemset that contains k-items) are used to explore (k+1)-itemsets. In the beginning, the set of frequent 1-itemsets is found. This set contains items that satisfy minimum support and is

denoted by L1. In each subsequent pass, we begin with a set of itemsets found to be frequent in the previous pass. This set is used for generating new itemsets, called candidate itemsets, and count the actual support for these candidate itemsets during the pass over the data. At the end of the pass, we determine which of the candidate itemsets are actually frequent and they are used in the next pass. Therefore, L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found. An important property called Apriori property is used to reduce the search space which is described as: "All nonempty subsets of a frequent itemset must also be frequent" [5]. How Lk-1 is used to find Lk is consisting of two steps, join and prune actions as followed:

1. **Join Step:** Join Lk-1 with itself to obtain the candidate itemset Ck.
2. **Prune Step:** Scan the database to determine the count of each candidate in Ck. When the count is less than the minimum support count, it should be delete from the candidate itemsets. Meanwhile, if any (k-1) subset of candidate k-itemset is not in Lk-1 then the candidate cannot be frequent either and so can be removed. After this, we get k-itemset which is denoted by Lk.

A. Limitations of Apriori Algorithm

Apriori algorithm, in spite of being simple and clear, has some limitations.

- It takes more time, space and memory for candidate generation process.
- To generate the candidate set, it requires multiple scan over the database. Repeatedly scanning the database requires a lot of I/O load [4].

In order to overcome the drawbacks of Apriori Algorithm, FP-Growth algorithm has been developed. The main difference between the two approaches is that the Apriori algorithm generates the candidate itemsets but FP-Growth does not generate the candidate itemset.

IV. FP-GROWTH ALGORITHM

FP-Growth algorithm is an efficient algorithm for producing the frequent itemsets without generation of candidate itemsets. It is based upon the divide and conquers strategy. It needs only two scans over the database for finding all frequent itemsets. This approach compresses the database representing frequent itemset into FP-tree. Then in the next 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 [6].

Particularly, constructing the FP-tree and generating frequent patterns from the FP-Tree are the main steps in FP-growth algorithm.

The process of constructing FP-Tree is as follows: First create root of the tree labelled with "null". Scan database second time as we scanned first time to create 1-itemset. Process items in each transaction in decreasing order of their frequency. A new branch is created for each transaction with the corresponding support. If the same node is encountered in another transaction, just increment the support count of common node. Each item points to the occurrence in the tree using the chain of node-link by maintaining the header table. Now the problem of mining frequent patterns in database is transformed to that of mining the FP-Tree. The constructed FP-tree is mined as:

1. Start from each frequent length-1 pattern (as an initial suffix pattern), construct its conditional pattern base.
2. Then, construct its conditional FP-Tree and perform mining on such a tree.
3. The pattern growth is achieved by concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-Tree.
4. The union of frequent pattern (generated by step 3) gives the required frequent itemset [7].

A. Advantages of FP-Growth Algorithm

The main advantages of FP-Growth algorithm are:

- FP-Tree is a compressed representation of the original database because only frequent items are used to construct the tree, other irrelevant information are pruned.
- This algorithm only scans the database twice.
- FP-Tree uses a divide and conquers method that considerably reduced the size of the subsequent conditional FP-Tree [3].

V. METHODOLOGY AND RESULTS

The two association rule mining algorithms (Apriori and FP-Growth) were implemented in WEKA. Weka is a collection of machine learning algorithms for data mining tasks. It is a java based software contains tools for data pre-processing, classification, regression, clustering, association rules.

The Supermarket dataset is used for the experimentation. This dataset contains 4627 instances and 217 attributes. The performance of Apriori and FP-Growth algorithms was evaluated based upon execution time and number of scans for different number of instances. The Table I shows the Execution time for both the Apriori and FP-Growth Algorithms for different number of instances.

TABLE I
EXECUTION TIME FOR DIFFERENT NUMBER OF INSTANCES

No. of Instances	Execution Time (Secs.)	
	Apriori	FP-Growth
991	8	1
1975	26	1
3415	49	2
4627	61	3

Fig. 1 shows that when the number of instances increases, execution time also increases. The Apriori and FP-Growth algorithms take 8 seconds and 1 second respectively when number of instances are 991. These results show that FP-Growth Algorithm takes less time than Apriori Algorithm.

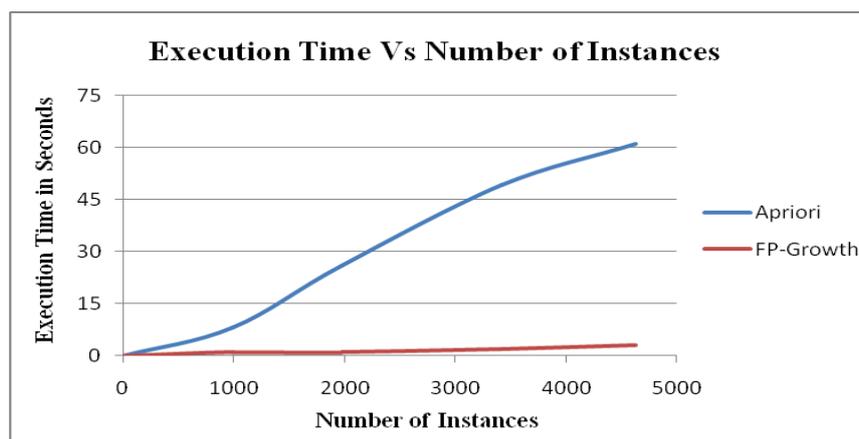


Fig. 1 Execution Time Vs Number of Instances

Fig. 2 shows that FP-Growth algorithm takes less scans than Apriori algorithm. So FP-Growth outperforms Apriori based on the number of scans for various numbers of instances.

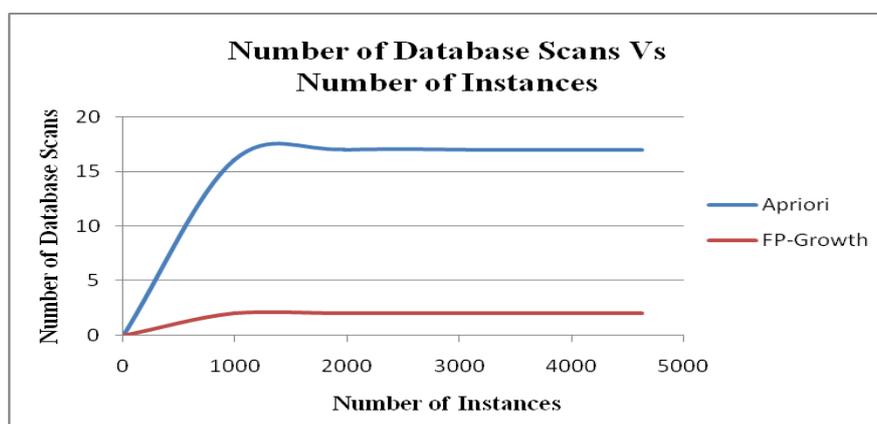


Fig. 2 Number of Database Scans Vs Number of Instances

VI. CONCLUSION

The association rules play a major role in many data mining applications, trying to find interesting patterns in databases. Various algorithms have been developed for mining association rules. The most common algorithms are the Apriori and FP-Growth algorithms. The performance of two algorithms is analyzed based upon execution time and number of scans for different number of instances. The results show that FP-Growth outperforms Apriori in terms of execution time and number of database scans.

REFERENCES

- [1] Venkatadri.M and Dr. Lokanatha C. Reddy, "A Review on Data mining from Past to the Future", International Journal of Computer Applications, Volume 15– No.7, pp. 19-22, February 2011.
- [2] Agrawal R., Imielinski T. and Swami, A. "Mining Association Rules between Sets of Items in Large Databases". In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207-216.
- [3] Sotiris Kotsiantis and Dimitris Kanellopoulos, "Association Rules Mining: A Recent Overview", International Transactions on Computer Science and Engineering, Volume 32 (1), pp. 71-82, 2006.
- [4] R.Divya and S.Vinod kumar, "Survey on AIS, Apriori And FP-Tree Algorithms", International Journal of Computer Science and Management Research, Volume 1, Issue 2, pp. 194- 200, September 2012.
- [5] Han J. and M. Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann publishers, 2nd Edition.
- [6] [Online]. Available: http://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_FP-Growth_Algorithm#FP-Tree_structure
- [7] Pinki Sharma and Rakesh Sharma, "Study of Mining Frequent Patterns at Various Levels of Abstraction", International Journal of Advanced Research in Computer Science, Volume 1, No. 2, pp. 197-201, July-August 2010.