Performance Evaluation of Apriori and FP-Growth Algorithms

M.S. Mythili,
Assistant Professor,
Bishop Heber College,Tiruchirappalli

A.R. Mohamed Shanavas, Ph.D
Associate Professor,
Jamal Mohamed College,
Tiruchirappalli

ABSTRACT
In Data Mining, Association Rule Mining is a standard and well researched technique for locating fascinating relations between variables in large databases. Association rule is used as a precursor to different Data Mining techniques like classification, clustering and prediction. The aim of the paper is to guage the performance of the Apriori algorithm and Frequent Pattern (FP) growth algorithm by comparing their capabilities. The evaluation study shows that the

FP-growth algorithm is efficient and ascendable than the Apriori algorithm.

General Terms
Data Mining, Association Rule Mining

Keywords
Apriori, FP-growth, Support, Confidence

1. INTRODUCTION
Data Mining could be a promising and flourishing frontier in analysis of data and also the results of analysis has several applications. Data Mining, also popularly referred as Knowledge Discovery from Data (KDD), is the machine-driven or convenient extraction of patterns representing knowledge implicitly keep or captured in huge databases, data warehouses, the Web, data repositories, and information streams.

Data Mining could be a multidisciplinary field, drawing from areas like information technology, machine learning, statistics, pattern recognition, data retrieval, neural networks; Knowledge based systems, artificial intelligence and data visualization.

Association Mining aims to extract attention-grabbing correlations, frequent patterns, and association structures among set of things or objects in transaction data based relational databases or different data repositories. Two statistical measures that govern Association Rule Mining are Support and Confidence. Support should be measured as to how often it should occur in the database. Confidence may well be gauged to seek out the strength of the rule. The Association rules are interesting if they satisfy each a minimum Support threshold and a minimum Confidence threshold [1].

This paper aims to present a performance evaluation of Apriori and FP-growth algorithms. The distinction between the two algorithms is that the Apriori algorithm generates candidate frequent itemsets and also the FP-growth algorithm avoids candidate generation and it develops a tree by economical and efficient 'divide and conquer' strategy.

2. ASSOCIATION RULE MINING
An Association rule is an expression of the form $X \rightarrow Y$ means that whenever X seems, Y also tends to appear. X and Y are itemsets. An itemsets is nothing but a collection of database items. X is usually stated as the rule's antecedent and Y as the consequent of the rule.

Association rules are stated as Boolean rules encompassing with Support and Confidence. Support is the proportion of transactions in an exceedingly information that satisfy the rule.

Confidence denotes the chance of Y being a true subject to X or P (Y|X).

Association Rule Mining is usually split up into two separate steps as stipulated below.

1. Find all frequent itemset:
   An itemset that happens a minimum as often as a planned minimum Support count.
2. Generate strong Association rules from the frequent Itemset:
   The rules should satisfy minimum Support and minimum Confidence.

This paper is organized as follows. Section 2 presents the idea of Association Rule Mining and discusses the aspects of Apriori and FP-growth algorithm. Section 3 elaborates a comparative analysis of Apriori algorithm and FP-growth algorithm. Section 4 explains the experimental results. Section 5 describes the results and discussions. Section 6 provides the conclusion.

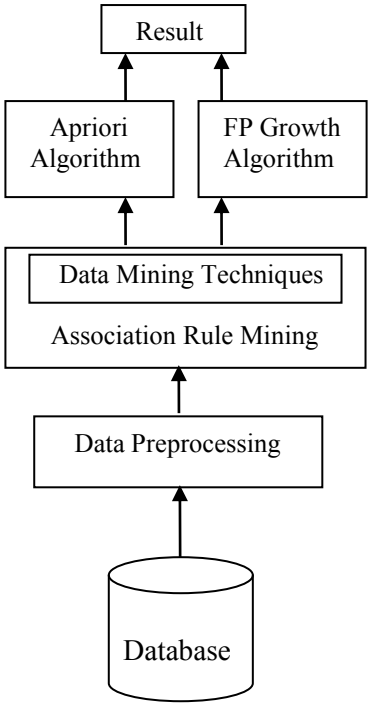

Figure 1: Association Rule Mining in Mining

2.1 Apriori Algorithm

One of the first and best algorithms for mining all frequent itemsets and Association Rule Mining was Apriori algorithm projected by Agrawal et al. in 1993[3].The idea of Apriori algorithmic program is to form multiple passes over the database. Apriori (level wise algorithm) relies on the

Anti-monotonic property of set theory states that every set of the frequent itemset is additionally frequent. Apriori could also be a candidate generation algorithmic program and issue in an extremely level wise fashion.It uses breadth first search and a tree structure to count candidate itemsets efficiently. The Apriori property follows two step processes:

- Join step: - C_k is generated by combining l_{k-1} with itself.
- Prune Step: - Any (k – 1) item set that's not frequent cannot be a set of a frequent k item set.

2.1.1 Apriori Algorithm Pseudocode

Procedure **Apriori** (T, *mSupport*) {//T is the database and *mSupport* is that the minimum Support
L_1 = {frequent items};
For (k= 2; L_{k-1}! =∅ ; k++) **{**
C_k= candidates generated from L_{k-1}
For each transaction **t** in database **do {**
Increment the count of all candidates in C_k that are contained in t
L_k = candidates in C_k with *mSupport*
}//end is for every statement
}//end is for
Return$U_k L_k$;
}

2.1.2 Benefits of Apriori Algorithm

- Large itemset could be utilized.
- It is extremely convenient and simple for implementation.

2.1.3 Drawbacks of Apriori Algorithm

- The Apriori algorithmic program takes longer time for candidate generation technique.
- The Apriori algorithmic program needs many scans
- of the database.
- Many trivial rules are derived and it will be hard to extract the most interesting rules.
- Rules can be inexplicable and fine grained.
- Redundant rules are generated.

2.2 FP Growth Algorithm

The Apriori algorithmic program based upon the Anti-monotonic property. The two main issues are, repeated database scan and high execution time. There is a need for compact data structure for mining frequent itemset.

FP growth algorithmic program is an efficient algorithm for producing the frequent itemsets without generation of candidate itemsets. It adopts a divide and conquer strategy and it needs two database scans to seek out the Support count. It can mine the items by using lift, leverage and conviction by specifying minimum threshold. [2]

2.2.1 Generating FP-Trees Pseudocode

The algorithmic program works as follows:

1. Scan the transaction database once, as among the Apriori algorithmic program, to seek out all the frequent items and their Support.
2. Sort the frequent items in descending order of their Support.
3. Initially, begin making the FP-tree with a root "null".
4. Get the primary transaction from the transaction database. Takeaway all non-frequent items and list the remaining items in line with the order among the sorted frequent items.
5. Use the transaction to construct the primary branch of the tree with each node corresponding to a frequent item and showing that item's frequency that's one for the primary transaction.
6. Get the next transaction from the transaction database. Takeaway all non-frequent items and list the remaining items in line with the order among the sorted frequent items.
7. Insert the transaction within the tree using any common prefix that may appear. Increase the item counts.
8. Continue with Step 6 until all transactions among the database are processed.

2.2.2 FP-Tree Algorithmic Approach

The FP-growth algorithmic program for mining frequent patterns with FP-tree by pattern fragment growth is:
Input: a FP-tree created with the above mentioned algorithm;

D – Transaction database;
S – Minimum Support threshold.
Output: The full set of frequent patterns.
Method: decision FP (FP-tree, null).
Procedure FP (Tree, A)
{
If Tree contains a one path P then for each combination (denoted as B) of the nodes among the trail P do generate

pattern B ∪ A with sup=minimum Support of nodes in B else
for each ai among the header of the Tree do
{
Generate pattern B = ai ∪ A with sup = ai.Support;
Construct B's cond pattern base and B's cond FP-tree
Tree B;
if Tree B≠θ
then decision FP(Tree B, B)
}
}

3 COMPARATIVE ANALYSES

Table 1 Apriori and FP-growth comparisons

S.No	Parameters	Apriori	FP-growth
1	Storage structure	Array based	Tree based
2	Search type	Breadth First Search	Divide and conquer
3	Technique	Join and prune	Constructs conditional frequency pattern tree which satisfy minimum Support
4	Number of Database scans	K+1 scans	2 scans
5	Memory utilization	Large memory (candidate generation)	Less memory (No candidate generation).
6	Database	Sparse/dense datasets	Large and medium data sets
7	Run time	More time	Less time

4. EXPERIMENTAL RESULTS

Datasets employed in the Apriori and FP-growth algorithm should be clear and will be preprocessed for handling missing or redundant attributes. The data are to be handled efficiently to get the best outcome from the Data Mining process.

For the experimental study two datasets will be used.
The datasets was obtained from the UCI repository of machine learning databases. The details of datasets are

Datasets Name	Number of Instances	Number of Attributes
Mushroom	8124	22
Super Market	4627	217

The two datasets were tested in RapidMiner4.1 software. The software package is an assortment of open source Data Mining and machine learning algorithms.

5. RESULTS AND DISCUSSIONS

As a results of the experimental study of the Mushroom and Super market datasets, it's clearly revealed that the performance of FP-growth algorithm is better than the Apriori algorithm. The execution time is that the time to mine the frequent itemsets with completely different transactions.

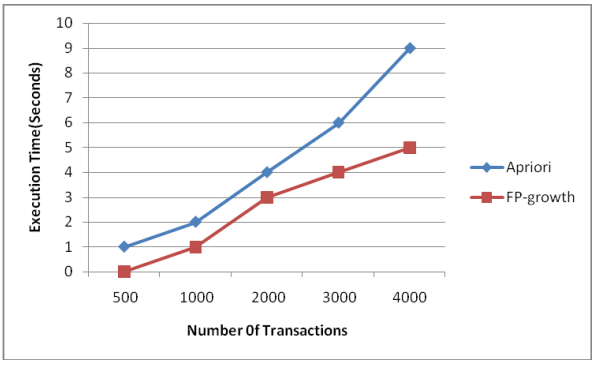

Figure 2: Execution Time with increasing Transactions for Mushroom data (support=0.1)

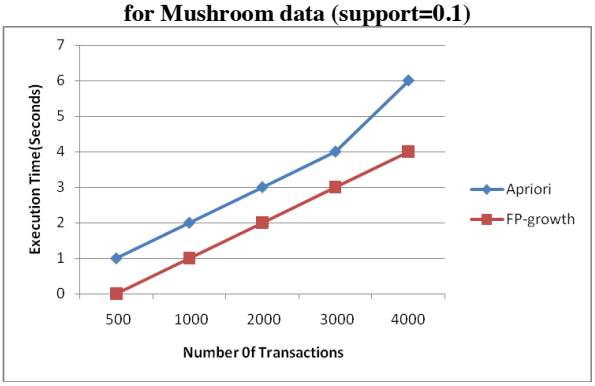

Figure 3: Execution Time with increasing Transactions for Mushroom data (support=0.5)

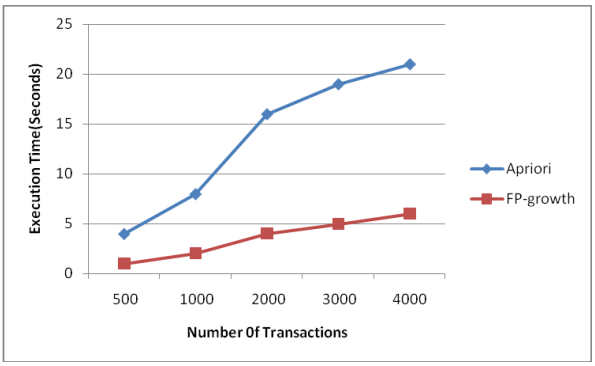

Figure4: Execution Time with increasing Transactions for Super Market data (support=0.1)

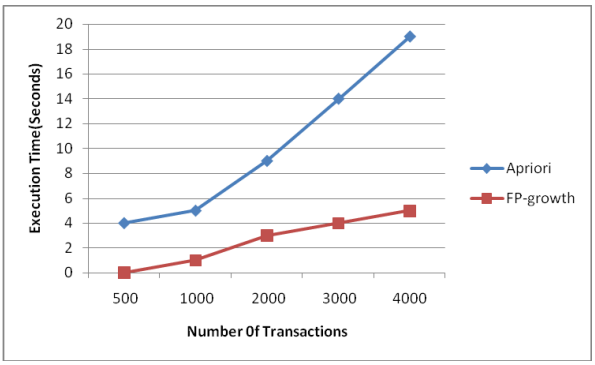

Figure 5: Execution Time with increasing Transactions for Super Market data (support=0.5)

Figure 2 to 5 shows that when the number of transactions Increases; the execution time for both the algorithms are

increased with the minimum support levels. The results show that the FP-growth algorithm requires only less execution time than the Apriori algorithm.

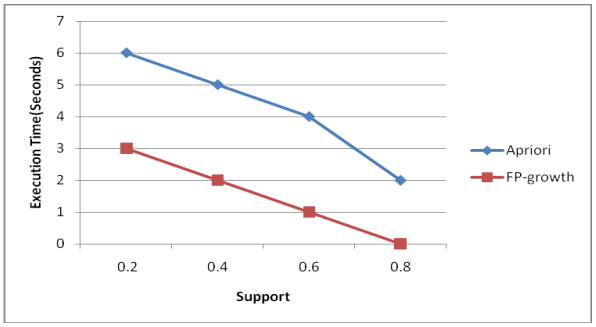

Figure 6: Execution Time with various Support levels for Mushroom data (2000 transaction)

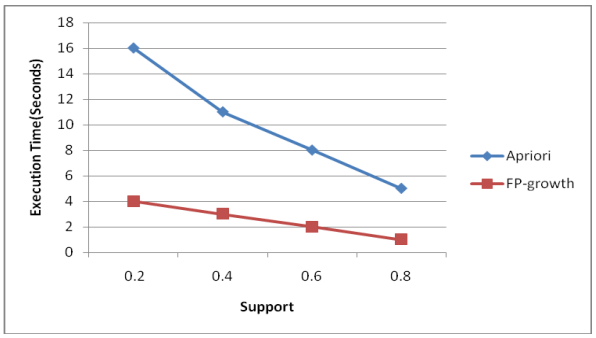

Figure 7: Execution Time with various Support levels for Mushroom data (4000 transaction)

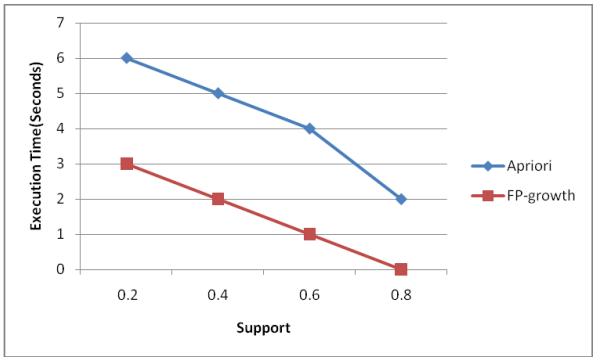

Figure 8: Execution Time with various Support levels for Super Market data (2000 transaction)

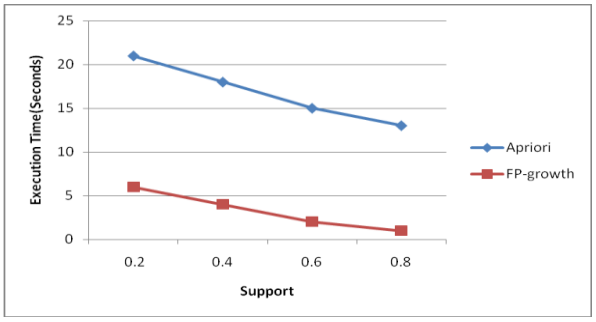

Figure 9: Execution Time with various Support levels for Super Market data (4000 transaction)

Figure 6 to 9 shows the performance analysis of run time of Apriori algorithm and FP-growth algorithm for various Support levels. It reveals that the time taken to execute the FP-growth algorithm is extremely less compared to Apriori algorithm for any Support level. The time of execution is decreased with the minimum support level.

6. CONCLUSION

It is concluded that Association Rule Mining is an interesting pattern mining problem. The algorithms used are conceptually clear and ensuing results are perceivable. From the experimental data conferred, it is concluded that the FP-growth algorithm performs better than the Apriori algorithm. In future, it is possible to extend the research by using the different clustering techniques and also the Association Rule Mining for large number of databases.

7. REFERENCES

[1] N.P. Gopalan and B. Sivaselvan, "Data Mining Techniques and Trends", PHI Learning private limited, New Delhi,2009.

[2] Han, J., Kamber, M.,"Data Mining concepts and techniques", Elsevier Inc., Second Edition, San Francisco, 2006.

[3] R. Agrawal, R. Srikant. "Fast algorithms for mining association rules in large databases". *Proc.of 20th Int'l conf. on VLDB*: 487-499, 1994.

[4] Ashok Savasere,Edward Omieinski and Shankant Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases", Proceedings of the 21st International Conference on Very Large Data Bases, pp. 432 – 444,2005.

[5] R.Agrawal, T.Imielinski, and A.Swami, "Mining Association Rules Between Sets Of Items In Large Databases", In proceedings of the ACM SIGMOD International Conference on Management of data, pp. 207-216,1993.

[6] M. J. Zaki and C.J. Hsiao, "CHARM: An efficient algorithm for closed association rule mining", Technical Report 99-10, Computer Science Dept., Rensselaer Polytechnic Institute,1999.

[7] Sotiris Kotsiantis and Dimitris Kanellopoulos,"Association Rules Mining: A Recent Overview", GESTS International Transactions on Computer Science and Engineering, Vol.32, No: 1, pp. 71-82, 2006.

[8] Anurag Choubey, Ravindra Patel,J. L.Rana, "A Survey Of Efficient Algorithms And New Approach For Fast Discovery Of Frequent Item Set For Association Rule Mining", International Journal of Soft Computing and Engineering,2011.

[9] Rakesh Agrawal and Ramakrishnan Srikant, "Fast Algorithms For Mining Association Rules In Large Databases", In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pp 487- 499, Santiago, Chile,1994.

[10] M. Houtsma, and Arun Swami, "Set-Oriented Mining for Association Rules in Relational Databases", IEEE International Conference on Data Engineering, pp. 25–33, 1995.