Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 2, February 2017, pg.81 – 89

PERFORMANCE COMPARISON OF APRIORI, ECLAT AND FP-GROWTH ALGORITHM FOR ASSOCIATION RULE LEARNING

GAYATHRI.G.S
[1]Computer Science and Engineering, VTU, India
gayathrigummaraj@gmail.com

Abstract*— The main aim is to generate a frequent itemset. Big Data analytics is the process of examining big data to uncover hidden patterns. Association Rule Learning is a technique which is used to implement big data. It finds the frequent items in the dataset. Frequent itemsets are those items which occur frequently in the database. To find the frequent itemsets, we are using three algorithms APRIORI ALGORITHM, ECLAT ALGORITHM and FP-GROWTH ALGORITHM. The performance of these three algorithms is compared based on the Time efficiency. After the comparison, we conclude that APRIORI algorithm is the fastest algorithm for large dataset and FP-GROWTH algorithm is the fastest algorithm for small dataset. It takes less time to generate the frequent item-sets as compared to other algorithm, that is, ECLAT algorithm.

***Keywords*— Big Data analytics, Association Rule Learning, frequent item-sets**

I. INTRODUCTION

Big Data is the application of specialized techniques and technologies to process very large sets of data. These data sets are often so large and complex that it becomes difficult to process using on-hand database management tools. Examples include web logs, call records, medical records, military surveillance, photography archives, video archives and large-scale e-commerce.

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. 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 term data mining or knowledge discovery in database has been adopted for a field of research dealing with the automatic discovery of implicit information or knowledge within the databases. The implicit information within databases, mainly the interesting association relationships among sets of objects that lead to association rules may disclose useful patterns for decision support, financial forecast, marketing policies, even medical diagnosis and many other applications.

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, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems. The original motivation for searching association rules came from the need to analyze so called supermarket transaction data, that is, to examine

customer behavior in terms of the purchased products. Association rules describe how often items are purchased together.

Association Rule Learning has become particularly popular among marketers. In fact, an example of association rule mining is known as market basket analysis. The task is to find which items are frequently purchased together. This knowledge can be used by professionals to plan layouts and to place items that are frequently bought together in close proximity to each other, thus helping to improve the sales. Association rule learning involves the relationships between items in a data set. Association rule mining classifies a given transaction as a subset of the set of all possible items. Association rule mining finds out item sets which have minimum support and are represented in a relatively high number of transactions. These transactions are simply known as frequent item sets. The algorithms that use association rules are divided into two stages, the first is to find the frequent sets and the second is to use these frequent sets to generate the association rules.

Apriori Algorithm is one of the most important algorithm which is used to extract frequent item-sets from large database and get the association rule for discovering the knowledge. It basically requires two important things: minimum support and minimum confidence. First, we check whether the items are greater than or equal to the minimum support and we find the frequent item-sets respectively. Secondly, the minimum confidence constraint is used to form association rules.

Eclat Algorithm finds the elements from bottom like depth first search. Eclat algorithm is very simple algorithm to find the frequent item sets. This algorithm uses vertical database. It cannot use horizontal database. If there is any horizontal database, then we need to convert into vertical database. There is no need to scan the database again and again. Eclat algorithm scans the database only once. Support is counted in this algorithm. Confidence is not calculated in this algorithm.

One of the algorithms that does not use any candidates to discover the frequent patterns is the **FP-Growth (Frequent Pattern Growth)** algorithm proposed. The other main difference to the Apriori algorithm is the number of the database readings. While the Apriori is a level-wise algorithm, the FP-growth is a two-phase method. It reads the database only twice and stores the database in a form of a tree in the main memory. The algorithm works as follows. During the first database scan the number of occurrences of each item is determined and the infrequent ones are discarded. Then the frequent items are ordered descending their support. During the second database scan the transactions are read and the frequent items of them are inserted into a so-called FP-tree structure. In this way the database is pruned and is compressed into the memory. The aim of using FP-tree is to store the transactions in such a way that discovering the patterns can be achieved efficiently.

II. PROBLEM DEFINITION

The problem consists of finding associations between items or item-sets in transactional data. The data could be retail sales in the form of customer transactions or any collection of sets of observations. Formally, the problem is stated as follows:

Let $I=\{i_1,i_2,\ldots i_m\}$ be a set of literals, called items where 'm' is considered the dimensionality of the problem. Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. A unique identifier TID is given to each transaction. A transaction T is said to contain X, a set of items in I, if $X \subseteq T$. An association rule is an implication of the form "$X \Rightarrow Y$", where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \varphi$. An item-set X is said to be large or frequent if its support s is greater or equal than a given minimum support threshold (σ). The rule $X \Rightarrow Y$ has a support (s)in the transaction set D if s% of the transactions in D contains $X \cup Y$. In other words, the support of the rule is the probability that X and Y hold together among all the possible presented cases.

The problem of discovering all association rules from a set of transactions D consists of generating the rules that have a support greater than a given threshold. These rules are called strong rules. This association-mining task can be broken into two steps:

A. *A step for finding all frequent k-item-sets known for its extreme I/O scan expense, and the massive computational costs, and*

B. *A straightforward step for generating strong rules. In this paper, we are mainly interested in the first step.*

III. ALGORITHMS DESCRIPTION AND WORK FLOW

The Apriori, FPGrowth and Elact algorithms are described with their work flows below

A. *Aprori Algorithm*

Apriori principle: **"If an item set is frequent, then all of its subsets must be frequent."**

Apriori is designed to operate on databases containing transactions. The Apriori principle gives the advantage as it **reduce the number of items being considered by only exploring the item sets whose support count is greater than the minimum support count**.

This algorithm needs to perform two basic steps, which are:

1. Join - self-join with previous frequent L_{k-1} item-set and create new candidate C_{k+1} item-set.
2. Prune - filter from the current candidate item-set whose subset is not frequent in previous step

Working of Apriori Algorithm

1. Scan the entire database and find out the Candidate 1-item-set (C_1) along with occurrence count i.e., number of times each item has appeared in the database.
2. Apply Pruning on (C_1).
3. Frequent-1-item-set (L_1) is determined based on their frequency from Candidate 1-item-set (C_1) after pruning.
4. Then this frequent-1-item-set (L_1) is used to form candidate-2-item-set (C_2).
5. This candidate-2-item-set (C_2) is used to determine frequent-2-item-set (L_2).
6. Repeat step 2 to 5 until C_K is null.

Figure 1: Apriori Algorithm

Algorithm : Apriori Algorithm

1. Initially: Ck: Candidate itemset of size k,
Lk : Frequent itemset of size k.
2. Procedure: APRIORI ALGORITHM
3. for (k = 1; Lk ! = ϕ ; k++) do begin
a) Ck+1 = Candidates generated from Lk;
b) Prune (Ck+1);
c) for each transaction t in database do increment the count of all candidates in (Ck+1) that are contained in t;
d) (Lk+1) = candidates in (Ck+1) with min-support;
4. end
5. return Ck Lk
6. end procedure

B. FP-Growth Algorithm

FP stands for frequent pattern. FP-Tree frequent pattern analysis is used in the development of association rule learning. It allows frequent itemset discovery without candidate itemset generation.

The FP-growth algorithm can be divided into two phases: the construction of FP-Tree and mining frequent patterns from FP-Tree. The construction of FP-Tree requires two scans on database. The first scan selects frequent items which are then sorted by frequency in descending order to form F-list. The second scan constructs FP-Tree.

First, the transactions are reordered according to F-list, with non-frequent items removed. Then reordered transactions are inserted into FP-Tree. Input of FP-Growth is FP-Tree and the minimum support count. FP-Growth traverses nodes in the FP-Tree from the least frequent item in F-list. While visiting each node, FP-Growth collects items on the path from the node to the root of the tree. Those items form the so called conditional pattern base of that item.

The conditional pattern base is a small database of patterns which co-occur with the item. Then FP-Growth creates small FP-Tree from the conditional pattern base and executes FP-Growth on the FP-Tree. The process is recursively iterated until no conditional pattern base can be generated.

Working of FP Growth
Step 1:
a. Build a compact data structure called the FP tree.
b. Built using two passes over the dataset.
Step 2:
Extract frequent itemsets directly from the FP tree.

FP tree is constructed in two passes:-

Pass 1:

1. Scan the database and find the support of each item.
2. Discard infrequent items.
3. Sort frequent items in decreasing order based on their frequency.
4. Use this order when building the FP tree, so common prefixes can be shared.

Pass 2:

1. Nodes correspond to items and have a counter.
2. Create the root of the tree labelled as null.
3. FP growth reads one transactions at a time and maps it to path.
4. Fixed order is used, so path can overlap when transactions share items.
5. Pointers are maintained between nodes containing the same item creating singly linked list.
6. The frequent items are extracted from the FP tree.

Algorithm for Growth

```
procedure FPGrowth(Tree, ∝)
if Tree contains a single path P then
    for each β = comb. of nodes in P do
      pattern = β ∪ ∝
      sup = min(sup of the nodes in β )
else
    for each a_i in the header of Tree do
       generatepattern = β ∪ ∝
       sup  = a_i.support
       construct β's conditional pattern base
       FPTree = construct β's conditional FP-tree
       If FPTree != 0 then
            FPGrowth(FPTree, β)
```

C. Elact Algorithm

Eclat algorithm reduces the access time. It finds the elements from bottom like depth first search. It is very simple algorithm to find the frequent item sets. This algorithm uses vertical database. It cannot use horizontal database. If there is any horizontal database, then we need to convert into vertical database. There is no need to scan the database again and again. Eclat algorithm scans the database only once. Support is counted in this algorithm. Confidence is not calculated in this algorithm. Eclat algorithm follows Apriori principle, which states that:

"If an itemset is frequent, then all of its subsets must be frequent."

This algorithm also needs to perform two basic steps, which are:

1. **Join** - self join with previous frequent L_{k-1} itemset and create new candidate C_{k+1} itemset.
2. **Prune** - filter from the current candidate itemset whose subset is not frequent in previous step.

Working of Elact Algorithm

1. Transform the horizontally formatted data to the vertical format by scanning the database once.
2. Get TID list for each item, the support count of an itemset is the length of the TID set of the itemset.
3. The frequent k- itemsets can be used to construct the candidate (k+1)-itemsets.
4. This process repeats, with k incremented by 1 each time, until no frequent items or no candidate itemsets can be found.

Algorithm:-

Input: $F_k = \{I1, I2, ..., In\}$ // cluster of frequent k-itemsets.
Output: Frequent l-itemsets, $l > k$.

Bottom-Up (F_k) {
1. for all Ii F_k
2. $F_k+1 = \Phi$;
3. for all $Ij \in F_k$, $i < j$
4. $N = Ii \cap Ij$;
5. if $N.sup \geq min\_sup$ then
6. $F_k+1 = F_k+1 \cup N$;
7. end
8. end
9. end
10. if $F_k+1 \neq \Phi$ then
11. Bottom-Up(F_k+1);
12. end
13. }

D. Comparison and Result analysis

The comparison of three algorithms of Association Rule Mining is done based on time efficiency. The comparison has been done on Apriori, FP-Growth and Eclat algorithm based on Time Efficiency. The best algorithm takes less time to generate the frequent item-set.

Based on the time taken to generate the frequent item-sets, we found that the execution time decreases as the minimum support increases. The overall goal of the frequent item set mining process helps to form the association rules for further use.

a. Comparison based on time efficiency

By comparing the three algorithms based on time efficiency, we found that FP-Growth takes less time to generate the frequent items-sets.
Therefore, FP-Growth algorithm is the best algorithm to generate the frequent item-sets for smaller dataset

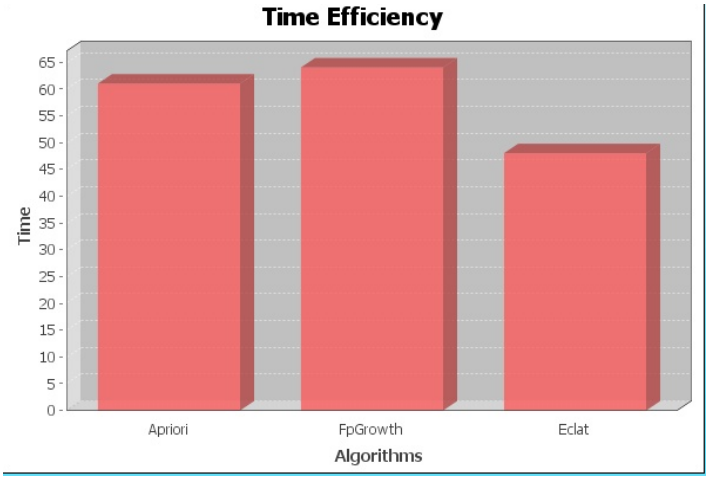

Fig-1 Comparison based on time efficiency

b. Comparison taking larger dataset as input

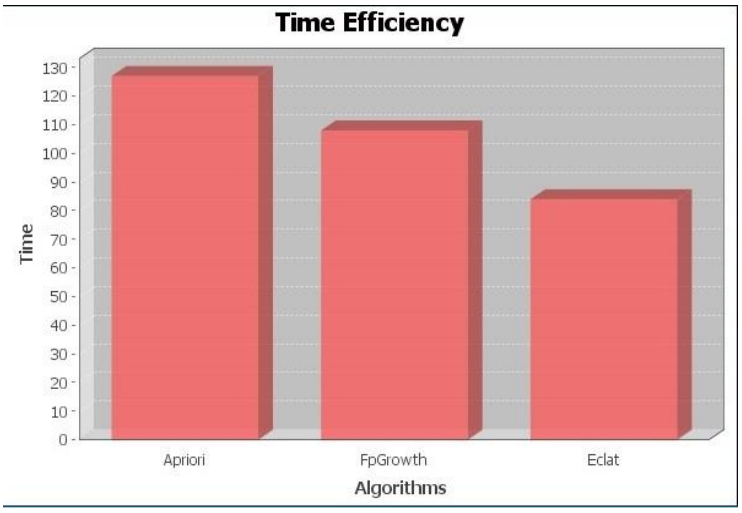

Fig-2 Comparison based on the size of a dataset

When we executed the three algorithms taking larger dataset as input, we found that, Apriori takes least time to generate the frequent item-set.

Therefore, we conclude that Apriori algorithm works faster on larger dataset whereas FP-Growth algorithm is more efficient for smaller dataset whereas Eclat algorithm takes almost same time to generate the frequent item-sets for both smaller and larger data-set.

IV. CONCLUSIONS AND FUTURE ENHANCEMENT

A. *Conclusion*

Here the three algorithms are compared to check for its efficiency. The algorithms are systemized and their performance is analyzed based on runtime and theoretical considerations. Despite the identified fundamental differences concerning employed strategies, runtime shown by algorithms is almost similar. The comparison shows that the Eclat algorithm outperforms the other two algorithms based on performance and time efficiency. We can make this claim based on the results of the graph generated on the online dataset.

It is also identified that the execution time decreases with increase in support. The above algorithms can be used in other domains to bring out interestingness among the data present in the repository. Association rules produced by these three algorithms can be combined to form efficient algorithms for better results for any real life application. Algorithms can also be combined to for an efficient algorithm.

B. *Future Enhancement*

ECLAT ALGORITHM

Improvement of Eclat Algorithm Based On Support In Frequent Itemset Mining

The new Bi-Eclat [19] algorithm sorted on support: Items sort in descending order according to the frequencies in transaction cache while itemsets use ascending order of support during support count. Compared with traditional Eclat algorithm, the results of experiments show that the Bi-Eclat algorithm gains better performance on several public databases given. Furthermore, the Bi-Eclat algorithm is applied in analyzing combination principles of prescriptions for Hepatitis B in Traditional Chinese Medicine, which shows its efficiency and effectiveness in practical usefulness.

APRIORI ALGORITHM

1. Organized Transaction Selection Approach

In this method the bottom-up approach is replaced by organized transaction selection approach. This algorithm uses organized transaction selection approach[10], where in the rules are generated by picking up the transactions according to the highest order first basis and hence avoiding generation of un-necessary patterns that are not a part of the original database

2. Apriori-Growth Algorithm

A new algorithm based on Apriori and the FP-tree structure is presented, which is called Apriori-Growth [11]. This method only scans the data set twice and builds FP-tree once while it still needs to generate candidate itemsets. The Apriori-Growth mainly includes two steps. First, the data set is scanned one time to find out the frequent 1 itemsets. Then the data set is scanned again to build an FP-tree. At last, the built FP-tree is mined by Apriori- Growth instead of FP-Growth

3. Apriori Algorithm Based On Infrequent Count

The proposed method [12] reduces CPU computation time by reducing transaction scan. The Concept infrequent count is based on minimum threshold support and 2- way searching to reduce execution time during scanning of transaction is introduced in proposed method. There exist several data mining algorithms for finding association rules but one of the candidate generation algorithms named Apriori algorithm is considered for the proposed work.

4. Efficient Way To Find Frequent Pattern Using Dynamic Programming Approach

In this paper [13], the improved candidate 1-itemsets generation and candidate 2-itemsets generation from traditional technique has been described. This algorithm utilizes the dynamic programming approachto facilitate fast candidate itemset generation and searching. The result of this algorithm is compared with the previous approach that optimizes the database scans and eliminates duplicate candidate itemset generation.

FP-GROWTH ALGORITHM

1. Web Usage Mining Using Improved Frequent Pattern Tree Algorithms

The aim of the proposed system [14] is to recognize usage pattern from web monitor files of a website. Apriori and FP Tree Algorithm is used for this. Both are prominent algorithms for mining frequent item sets for Boolean association rules. In computer science and data mining, Apriori is a typical algorithm for understand association rules [15]. Apriori Algorithm follows "bottom-up" technique, used to design to operate on databases containing transactions.

2. Incremental FP-Growth Mining Strategy For Dynamic Threshold Value And Database Based On MapReduce

The paper [16] presents a parallelized incremental FP-Growth mining strategy based on MapReduce, which aims to process large-scale data. The proposed incremental algorithm realizes effective data mining when threshold value and original database change at the same time. This novel algorithm is implemented on Hadoop and shows great advantages according to the experiment l results.

3. Balanced Parallel FP-Growth With MapReduce

In this paper [17], we propose a balanced parallel FP-Growth algorithm BPFP, based on the PFP algorithm, which parallelizes FP-Growth in the MapReduce approach. BPFP adds into PFP load balance feature, which improves parallelization and thereby improves performance. Through empirical study, BPFP outperformed the PFP[18] which uses some simple grouping strategy.

ACKNOWLEDGEMENT

I would like to express my gratitude to my guide **Dr.Prashant.CSR**, HOD and Dean Academics ,NHCE for his constant support, encouragement.
Also i thank the management, **Dr. Mohan Manghnani**, Chairman of NEW HORIZON EDUCTIONAL INSTITUTIONS for providing necessary infrastructure and creating good environment.

REFERENCES

[1] Ms. VibhavariChavan, Prof. Rajesh. N. Phursule, "Survey Paper On Big Data", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6), 2014.

[2] Chun-Wei Tsai, Chin-Feng Lai, Han-Chieh Chao and Athanasios V. Vasilakos, "Big data analytics: A Survey", Journal of Big Data 2015.

[3] VarshaMashoria, Anju Singh, "Literature Survey on Various Frequent Pattern Mining Algorithm", IOSR Journal of Engineering (IOSRJEN), Vol. 3, Issue 1 (Jan. 2013), PP 58-64.

[4] VarshaMashoria, Anju Singh, "Literature Survey on Various Frequent Pattern Mining Algorithm", IOSR Journal of Engineering (IOSRJEN), Vol. 3, Issue 1 (Jan. 2013), PP 58-64.

[5] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo, "Fast discovery of association rules".
In U.M., Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, "Advances in Knowledge Discovery and Data Mining", pages 307–328. MIT Press, 1996.

[6] Arpan Shah, Pratik A. Patel, "A Collaborative Approach of Frequent Item Set Mining: A Survey", International Journal of Computer Applications (0975 – 8887), Volume 107 – No 8, December 2014.

[7] S. Neelima, N. Satyanarayana and P. Krishna Murthy3, "A Survey on Approaches for Mining Frequent Itemsets", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-87.

[8] ManishaGirotra, KanikaNagpal, SaloniMinocha, Neha Sharma, "Comparative Survey on Association Rule Mining Algorithms", International Journal of Computer Applications (0975 – 8887), Volume 84 – No 10, December 2013.

[9] Divya R., and Kumar, V.S. "Survey on AIS, Apriori and FP-Tree algorithm". International Journal of Computer Science and Management Research.Volume 1 Issue 2 September-2012 ISSN 2278-733X.

[10] K.R.Suneetha, R.Krishnamoorti, "Advanced Version of Apriori Algorithm", First International Conference on Integrated Intelligent Computing-2010, International Journal of Emerging Trends in Engineering and Development, Issue 5, Vol. 3 (April.-May. 2015).

[11] Bo Wu, Defu Zhang, QihuaLan, JieminZheng, "An Efficient Frequent Patterns Mining Algorithm based on Apriori Algorithm and the FP-tree Structure", China-Third 2008 International Conference on Convergence and Hybrid Information Technology-2008 IEEE.

[12] Shyam Kumar Singh, Preetham Kumar, "I2Apriori: An Improved Apriori Algorithm based on Infrequent Count", International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016978-1-4673-9939-5/16/©2016 IEEE.

[13] DharmeshBhalodiya, Chayya Patel, KM patel, "An efficient way to find frequent pattern with dynamic programing approach", Nima University International Conference on Engineering (NUiCONE) – 2013.

[14] Ashika Gupta, Ranjanasikarwar, Rakhiarora, NehaSaxena, "Web Usage Mining Using Improved Frequent Pattern Tree Algorithms", International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 978-1-4799-2900-9/14/©2014 IEEE.

[15] E-H. Han, G. Caryopsis "Scalable Data web mining for Association web Rules," IEEE Trans. Eng., vol. 12, no. 3, July 2012.

[16] Xiaoting Wei, Yunlong Ma ∗, Feng Zhang, Min Liu, WeimingShen, "Incremental FP-Growth Mining Strategy for Dynamic Threshold Value and Database Based on MapReduce", Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design.

[17] Le Zhou, ZhiyongZhong, Jin Chang, Junjie Li, Joshua Zhexue Huang, ShengzhongFeng, "Balanced Parallel FP-Growth with MapReduce", Center for High Performance Computing Institute of Advanced Computing and Digital Engineering Shenzhen Institutes of Advanced Technologies, Chinese Academy of Sciences- 978-1-4244-8886-5/10/©2010 IEEE.

[18] Haoyuan Li, Yi Wang, Dong Zhang, Ming Zhang, and Edward Y. Chang, "PFP: Parallel FP-Growth for Query", In Proceedings of the 2008 ACM conference on Recommender systems.

[19] Xiaomei Yu, Hong Wang, "Improvement of Eclat Algorithm Based on Support in Frequent Itemset Mining", JOURNAL OF COMPUTERS, VOL. 9, NO. 9, SEPTEMBER 2014.

