Advances in Intelligent Systems Research, volume 172
Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)

The Comparison of Apriori Algorithm with Preprocessing and FP-Growth Algorithm for Finding Frequent Data Pattern in Association Rule

Deo WICAKSONO[1], Muhammad Ihsan JAMBAK[2,*], and Danny Matthew SAPUTRA[1]

[1]*Informatics Department, Sriwijaya University, Palembang, Indonesia*
[2]*Informatics Management Department, Sriwijaya University, Palembang, Indonesia*
**Corresponding author: jambak@unsri.ac.id*

ABSTRACT
Association Rules is a data mining method to find the relation between items called rules. Finding rules in the association method can be divided into two phases. The first phase is finding the frequent pattern which satisfies specified minimum frequent, and the second phase is finding strict rules from the frequent pattern which satisfy the minimum support and confidence. The main problem of Association Rules is based on the algorithm used, and this method takes a large amount of memory and time-consuming. This study aims to add preprocessing using the aggregate function on the Apriori Algorithm and therefore improve the memory and time consumption for finding a large number of rules.
Keywords: *data mining, Association Rules, FP-Growth, apriori, rules*

Introduction

Association Rules is one of the Data Mining methods about mining frequent patterns from large databases to find the relations between data patterns, which is called frequent itemset that going to be used to find the rule. Rules are frequent itemset, which satisfies the specified minimum support and minimum confidence; support is the percentage of an itemset in the database, while confidence is how strong the relationship between the item in association rules. Association Rules use an algorithm to do its processes, such as Apriori and FP-Growth Algorithm.

Apriori Algorithm is one of the traditional and simple algorithms. Apriori algorithm using a Brute-force strategy to find data patterns by scanning the database repeatedly. The advantages of the Apriori algorithm is easy to implement and to study because the data structure is straightforward [1]. However, the disadvantage of the Apriori algorithm is the algorithm needs to scan the database repeatedly to generate candidate, this process takes a lot of time and memory, especially if the pattern it too many and long [2].

FP-Growth Algorithm using a root-like data structure and divide and conquer strategy to find candidate, this makes the FP-Growth algorithm as an efficient algorithm to find rules [3]. FP-Growth Algorithm can reduce memory and time used to find association rules because the FP-Growth algorithm only needs to scan the database two times to find rules candidates [4]. The disadvantage of the FP-Growth algorithm is, if the data is too long, the complexity of the data structure can reduce the performance [1].

However, with the disadvantages of the Apriori Algorithm, it does not mean the FP-Growth algorithm is superior compared to the Apriori Algorithm. Apriori Algorithm works better with a big dataset, while the FP-Growth Algorithm works better with a small dataset [1]. One of the advantages of the apriori algorithm is easy to learn and makes the apriori algorithm easy to use and developed by the researcher [5]. The Preprocess that was going to be done for Apriori in this research is to make the aggregate function to group the database. Because the Apriori algorithm needs to scan the database repeatedly, the preprocess aim to reduce the database every time it needs to scan so the time and memory used can be reduced.

Apriori Algorithm

The Apriori algorithm is one of the most basic and popular algorithms for association rules mining. Agrawal and Srikant proposed the Apriori algorithm in 1994. Until

now, this algorithm is the most used and developed by the researcher [5]. Apriori is designed to operate on databases containing transactions. In the Apriori algorithm, every transaction is seen as itemsets, with a given threshold, the algorithm will identify the item, which is subsets at least by minimum threshold as new itemsets.

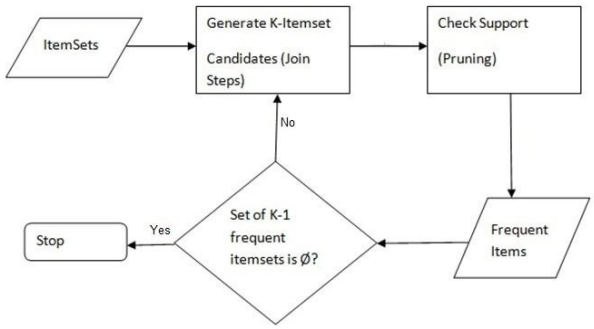

Figure 1. Flowchart of Apriori Algorithm

Apriori algorithm uses a "bottom-up" approach, where the itemsets are determined one item at a time; these steps known as candidate generation. This algorithm uses a breadth-first search and hash tree structure to count candidate itemsets efficiently. A group of candidates is tested against the data, which will be pruned if the candidates have an infrequent subpattern. This process repeats until no further successful extensions are found. Apriori algorithm is considered as a brute-force method because this method considers every k-itemset as the candidate of frequent itemset [6]. Consider the computational needed for every candidate is O(k), as a whole algorithm, the complexity of this method is $O(d*2^{d-1})$ [6].

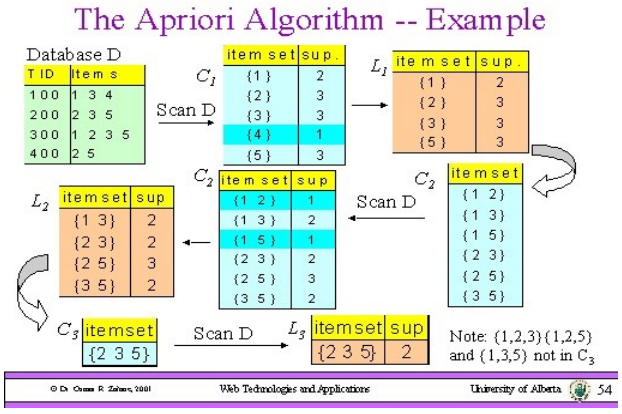

Figure 2. Example of how the Apriori Algorithm works. [9]

Because of the simple process of this algorithm, the advantage of this algorithm is more comfortable to learn, understand, and implemented; this is the reason this algorithm is called the most basic algorithm for association rules. However, this leads to its disadvantage, candidate generations, which is the primary process of the algorithm itself, is computationally costly. As the algorithm scan databases each time its try to generate new candidate, this process takes a lot of memory and processing time.

FP-GROWTH ALGORITHM

FP-Growth Algorithm is an efficient and scalable method for mining a complete set of frequent patterns by pattern fragment growth. FP-Growth algorithm was proposed by Han in 2000, using extended prefix tree structure for storing compressed information about frequent patterns named frequent-pattern tree. In his study proving that this method outperforms other method for frequent mining patterns.

FP-Growth algorithm is an efficient algorithm for association rules. In this algorithm, using an alternative way to find frequent itemsets without candidate generations which take a lot of memory and process time, this makes this algorithm performance is better than Apriori. As an alternative way, this algorithm uses a divide-and-conquer strategy and data structure called frequent-pattern tree to store frequent pattern mining.

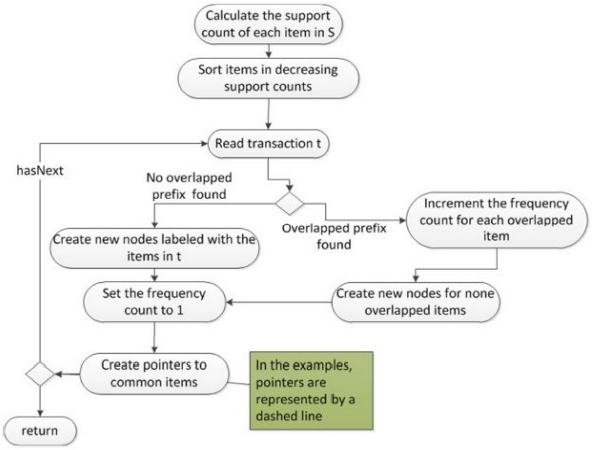

Figure 3. Flowchart of FP-Growth Algorithm

The run-time of the FP-Growth Algorithm is dependent on the compaction factor from the dataset [6]. If conditional FP-Tree of the result has many branches, then the algorithm performance will be drop drastically because the algorithm has to process a lot of conditional FP-Tree. The complexity of FP-Growth is very dependent on the pathfinding in FP-Tree for every item in the header table, which is dependent on how deep the tree is. The maximum depth of a tree is restricted by d for every conditional FP-Tree. Then the complexity of the algorithm is O (the amount of unique item in header table * the maximum depth of the tree) = O (d*d) [6].

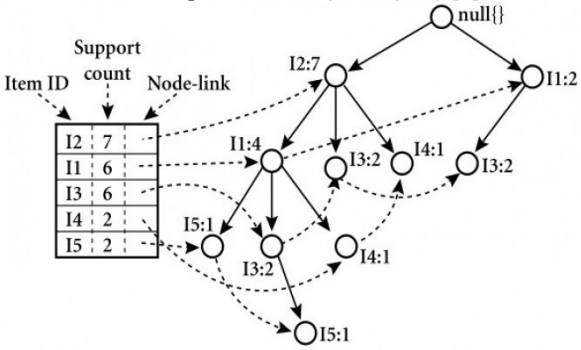

Figure 4. Example of FP-Tree [10]

Compared to the Apriori Algorithm, the FP-Growth algorithm is more complex to implement and understand.

However, on the other hand, this algorithm is more efficient. Therefore, the FP-Growth algorithm has much potential to develop and optimize.

MEASUREMENT METRIC

Measurement metrics on association rules are to determine if the generated rules are strict rules or useless rules. There are three metrics used in this research: Support, Confidence, and Dependency Factor.

1) Support

Support is the percentage of a transaction that has this rule [7]. That is mean, support is a value that indicates how frequent the itemset from the whole transaction.

$$P(X) = Support(X) = \frac{Frequency(X)}{Total\ Transactions} \quad (1)$$

$$Support(X \rightarrow Y) = Support(X \cup Y) \quad (2)$$

2) Confidence

Confidence is a value that determines how frequent the data pattern appears in frequent itemsets as a rule. Confidence is that association rules are to gauge how accurate a rule.

$$Confidences(X \rightarrow Y) = \frac{Frequency(X, Y)}{Frequency(X)} \quad (3)$$

3) Dependency Factor

Dependency is a metric that modified from certainty factor, and the difference is the dependency factor is based on real maximum and minimum values of conf(X→Y) for given value Support (X) and Support (Y) [8]. Also, the dependency factor determines by how much the value of the lift of a rule X→Y differs from the value one concerning how much it could have been different. It ranges within [-1,1]

$$DF(X \rightarrow Y) = \left\{ \frac{Conf(X \rightarrow Y) - P(Y)}{MaxConf(X \rightarrow Y | P(X), P(X)¿} - P \right. \quad (4)$$

Proposed Preprocessing on Apriori Algorithm

In this research, the proposed preprocessing on the Apriori Algorithm will be the aggregate function to group count how many items are transacted in a single transaction. Apriori algorithm scans the database repeatedly, so the time needed will increase as the database get larger. The database reduction will be made by comparing the k-itemset with the count of items in a single ID, and if the count is lower than k, then the ID will be ignored. There are the following steps of the proposed preprocessing:

Scan the database to determine the count of the item in a single transaction

When the k-itemset candidate generated, record the k

If Item count of an ID is less than k, ignore the transaction.

Figure 5. The proposed method of preprocessing

From Figure 5 can be explained as follows, first scan the database to create a table (i) which show how many item transacted in single ID, when finding k-itemset candidate, ignore ID that has lower Item count than k, for example in Figure 5 (iii) when finding 3-itemset, the ID 2,4 and 5 ignored from scanning (in red background) because its item count lower than k, which is 3. This preprocessing applied because k-itemset will not be found in ID, which contains items lower than k. With this preprocessing, then database scanned will be decreasing for every k-itemset.

EXPERIMENT

The dataset used for an experiment on this research is using a total of 1000 Transaction ID from CV. Sukses Inti Prima, dated July 2016 to December 2016. The dataset will be run with 3 Algorithm: Apriori Algorithm (A), FP-Growth Algorithm (FP), and Apriori Algorithm with preprocessing (AP). The dataset then experimented with four different scenarios. From Figure 6 to 9 show, by using different scenario, each algorithm has it is characteristic and advantages compared to another algorithm:

Using different Minimum Frequency

Using different Minimum Support

Using different Minimum Confidence

Using the different amounts of the dataset.

On FP-Growth Algorithm can be considered as the superior algorithm for s small amount of dataset. For a small dataset and rules. The memory and time usage is lower than Apriori and Apriori with preprocessing because of the tree structure data. However, for a large amount of dataset and rules, FP-Growth algorithm time and memory usage getting worse, the tree structure getting more complex.

On Apriori Algorithm can be considered as the superior algorithm for a large amount of dataset. Because to generate itemsets in Apriori Algorithm need to scan repeatedly makes this algorithm not right for finding a

small number of rules. However, because of the simple data structure, this algorithm performs better to find rules in a large dataset.

On Apriori Algorithm with preprocessing, generally need more time and memory compared to Apriori and FP-Growth Algorithm to find a small number of rules, because the algorithm needs more time and memory to do preprocessing compared to the optimized time and memory for finding the rules itself. However, for finding a large number of rules, this Algorithm performs better than the other two, as can be seen in Figure 5, and Figure 6, which visualizes the Apriori Algorithm with preprocessing perform, became better as the threshold lowered (lower threshold mean more rules generated).

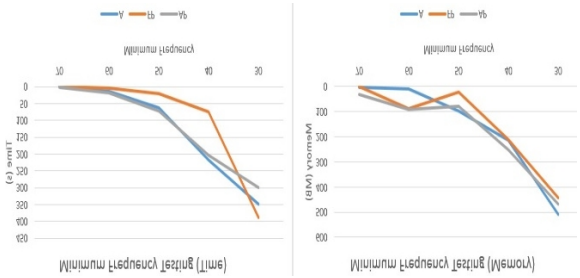

Figure 6. First Scenario, Different set of minimum frequency

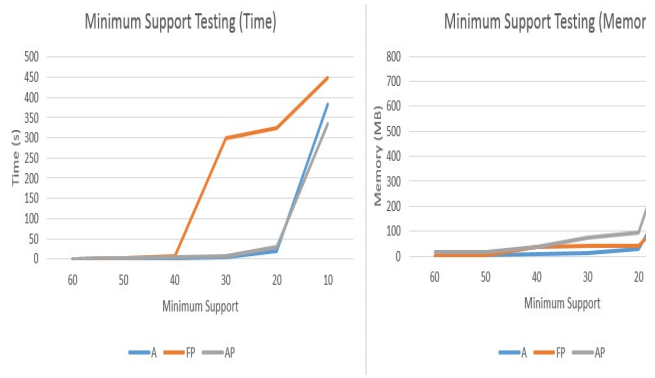

Figure 7. Second Scenario, Different set of minimum support

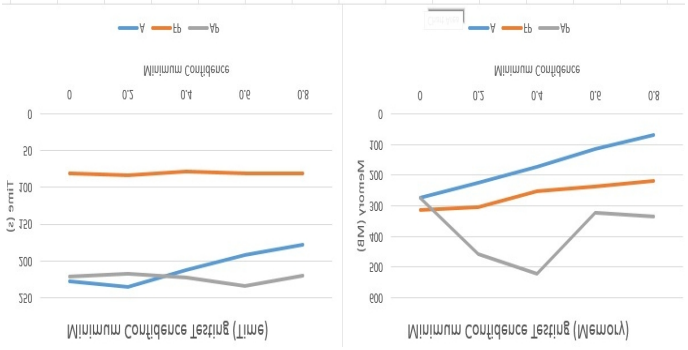

Figure 8. Third Scenario, Different set of Minimum Confidence

Figure 9. Fourth Scenario, Different set of Total ID

SUMMARY

This paper described a preprocessing stage that can be used for the implementation of the Apriori Algorithm, which contains how to aggregate function to grouping data to reduce the amount of time used on scanning data. As the experimental result show, each algorithm has its advantages for a different amount of dataset and rules generated. For the FP-Growth algorithm, it is superior for a small amount of dataset and rules; for the Apriori algorithm, its superior for a large amount of dataset, while for the Apriori algorithm with preprocessing superior for finding a large number of rules.

REFERENCES

[1] Kavitha, M., & Selvi, M. S. T. T., Comparative Study on Apriori Algorithm and Fp Growth Algorithm with Pros and Cons, *4*(4), 161–164. (2016)

[2] Shweta, M., & Garg, K., Mining Efficient Association Rules through Apriori Algorithm Using Attributes and Comparative Analysis of Various Association Rule Algorithms. International Journal of Advanced Research in Computer Science and Software Engineering, 3(6), 306–312. (2013)

[3] Borgelt, C., An Implementation of the FP-Growth Algorithm. , pp.759–770. (2005)

[4] Han, J., Pei, J. & Yin, Y., Mining frequent patterns without candidate generation. Proceedings of the 2000 ACM SIGMOD international conference on Management of Data - SIGMOD '00, pp.1–12. (2000)

[5] Anggraeni, R. M., "Perbandingan Algoritma Apriori dan Algoritma FP-

Growth Untuk Perekomendasi Pada Transaksi Peminjaman Buku di Perpustakaan Universitas Dian Nuswantoro." *Teknik Informatika, 1-5.* (2014)

[6] Tan, Pang-Ning, Michael Steinbach, Anuj Karpatne, and Vipin Kumar, *Introduction to Data Mining.* pp.327-414. (2018)

[7] Berzal, F. et al., Measuring the accuracy and interest of association rules: A new framework. *Intelligent Data Analysis*, 6, pp.221–235. (2002)

[8] Kryszkiewicz, M., Dependence Factor for Association Rules. , pp.135–145. (2015)

[9] Zaiane, Osmar R.,Principles of Knowledge Discovery in Databases. (1999)

[10] Chee, C.-H., Jaafar, J., Aziz, I. A., Hasan, M. H., & Yeoh, W., *Algorithms for frequent itemset mining: a literature review. Artificial Intelligence Review.* (2018)