International Journal of Computer Trends and Technology
ISSN: 2231-2803 / https://doi.org/10.14445/22312803/IJCTT-V67I3P101
Volume 67 Issue 3, 1- 4, March 2019

Original Article

Pattern Discovery with Web usage Mining using Apriori and FP-Growth Algorithms

Kondi Srujan Kumarr[1], M Ashish Naidu[2], K Radha[3]

[1, 2,3] *III-B.TECH-CSE, Rudraram, GITAM University, Hyderabad, Telangana, India*

Abstract - *In Data Mining, Association Rule Mining is a standard and well-researched technique for finding the relations between variables in large data sets. Association rule is used as a precursor to different Data Mining techniques like classification, clustering, and prediction. The paper aims to compare the performance of the Apriori algorithm and Frequent Pattern growth algorithm by comparing their capabilities and the Pros and cons of Apriori and FP-Growth Algorithms. The evaluation study shows that the FP-growth algorithm is more efficient than the Apriori algorithm. This paper presents the Pattern discovery from weblog data using web usage mining,* a Top-down approach to frequent mining itemsets.

Keywords *- Apriori, FP Growth, Classification, Prediction*

I. INTRODUCTION

Association Mining aims to extract correlations, frequent patterns, and association structures that are attention-grabbing among a 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 a minimum Support threshold and a minimum confidence threshold [2].

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 frequent candidate itemsets. The FP-growth algorithm avoids candidate generation. It develops a tree by economical and efficient 'divide and conquers strategy.

A. Data for Association Rules

Association models are designed to use transactional data. Nulls in transactional data are assumed to represent values that are known but not present in the transaction. For example, three out of hundreds of possible items might be purchased in a single transaction. The items that were not purchased are known but not present in the transaction. Transactional data, by its nature, is sparse. Only a small fraction of the attributes are non-zero or non-null in any given row. Apriori interprets all null values as indications of sparsity.

B. 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 item set is nothing but a collection of database items. X is usually stated as the rule's antecedent and Y as the rule's consequent. Association rules are stated as Boolean rules encompassing Support and Confidence. Support is the proportion of transactions a piece of exceeding information that satisfies the rule. Confidence denotes the chance of Y being a true subject to X or P (Y|X).
Association Rule Mining is usually split into two 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.

Advantages of FP-Growth Algorithm

These are the pros of Fp-growth:
There are only 2 passes over data-set

- This algorithm Compresses data-set
- There is no candidate generation.
- It is much faster than apriori.

Advantages of Apriori Algorithm

These are the pros of the apriori algorithm:
It is easy to understand and implement.

- These are the cons of the Apriori algorithm.
- When you need a large number of candidate rules, then this algorithm is computationally expensive.
- It is also expensive to calculate support because the calculation has to go through the entire database.

II. APRIORI ALGORITHM

Apriori algorithm, a classic algorithm, is useful in frequent mining itemsets and relevant association rules. Usually, you operate this algorithm on a database containing many transactions. One such example is the items customers buy at a supermarket. It helps the customers buy their items with ease and enhances the sales performance of the departmental store. This algorithm has utility in healthcare as it can help detect adverse drug reactions (ADR) by producing association rules to indicate the combination of medications and patient characteristics that could lead to ADRs [3]. Three significant components comprise the apriori algorithm. They are as follows.

- Support
- Confidence
- Lift

SUPPORT: Support is the ratio of transactions that includes all the items in the antecedent and consequent to the number of total transactions. Support can be expressed in probability notation as follows.

Support (A implies B) = P (A,B)

CONFIDENCE: Confidence is the ratio of the rule support to the number of transactions that include the antecedent.

Confidence can be expressed in probability notation as follows.

confidence (A implies B) = P (B/A) , which is equal to P(A,B) / P(A)

LIFT: Lift indicates the strength of a rule over the random co-occurrence of the antecedent and the consequent, given their support. It provides information about the improvement and the increase in the consequent probability given the antecedent. Lift is defined as follows.

(Rule Support) / (Support(Antecedent) * Support (Consequent))

A. Steps For Apriori Algorithm

Step1: Scan the transaction database to get the support S of each 1-itemset, compare S with min_sup, and get a set of frequent 1-itemsets, L1

Step2: Use L k-1 to join L k-1 to generate a set of candidate k-itemsets. And use Apriori property to prune the unfrequented k-itemsets from this set.

Step3: Scan the transaction database to get the support S of each candidate k-itemset in the final set, compare S with min_sup, and get a set of frequent k-itemsets, L k

Step4: The candidate set = Null N

If candidate set = Null, then repeat step 2

Step5: For each frequent itemset l, generate all nonempty subsets of l

Step6 : For every nonempty subset s of l, output the rule " s => (l-s)" if confidence C of the rule " s => (l-s)" (=support S of l/support S of s) ³ min_conf

The following methods can be used to improve the efficiency of the apriori algorithm

.**Transaction reduction** – A transaction not containing any frequent k-itemset becomes useless in subsequent scans.

Hash-based Itemset Counting – Exclude the k-itemset whose corresponding hashing bucket count is less than the threshold is an infrequent itemset.

B. Fp Growth Algorithm

Fp growth algorithm (frequent pattern growth). FP growth algorithm is an improvement of the apriori algorithm. FP growth algorithm used for finding frequent itemset in a transaction database without candidate generation. FP growth represents frequent items in frequent pattern trees or FP-tree.

FP-Tree is constructed using 2 passes over the data-set:

Pass 1 :

- Scan data and find support for each item
- Discard infrequent items
- Sort frequent items in decreasing order based on their support.

Use this order when building the FP-Tree, so common prefixes can be shared.

Pass 2 :

Nodes correspond to items and have a counter

- FP-Growth reads 1 transaction at a time and maps it to a path
- Fixed order is used, so paths can overlap when transactions share items (when they have the same prefix)
- In this case, counters are incremented

Pointers are maintained between nodes containing the same item, creating singly-linked lists (dotted lines)

- The more paths that overlap, the higher the compression. FP-tree may fit in the memory.

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 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. Take away 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. Take away 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 in the database are processed.

C. Discrimination Of Apriori V/S Fp-Growth

Various Comparisons are explained below with the help of different parameters for Apriori and FP-Growth Algorithms [1], [2].

Table 1. Comparisons Of Apriori And Fp-Growth Algorithms

SNO	Parameter	Apriori	FP-Growth
1	Technique	Generates singletons ,pairs ,triplets ,etc	Insert sorted items by frequency into a pattern tree
2	Runtime	Candidate generation is extremely slow. The runtime increases exponentially depending on the number of different items	Runtime increases linearly, depending on the number of transactions and items
3	Memory Usage	Saves singletons ,pairs ,triplets ,etc.	Stores a compact version of the database
4	Parallelizability	Candidate generation is very parallelizable	Data are very interdependent, and each node needs the root
5	Search type	Breadth-first search	Divide and conquer
6	Database	Sparse /dense datasets	Large and medium data-sets

III.PATTERN DISCOVERY FROM WEBLOG

Web usage mining is the mining method used for user browsing and access patterns [4]. At the side of the website, weblog mining is used to identify the web users to capture the data along with their browsing behaviors. This paper mainly aims to describe user behavior in classifying the patterns of web users' browsing and navigation data and also measure the performance of the Frequent Pattern Growth algorithm and Apriori algorithm by comparing their performances.

The Apriori algorithm and FP Growth algorithm are compared by applying the rapid miner tool to discover frequent user patterns and user behavior in the weblog. Both the algorithms help analyze the patterns of website usage and the features of user behavior knowledge obtained from web usage. For more effective browsing, personalization, and enhancing the web design, fp growth can be used. This experiment mainly focuses on the instances and time taken for execution calculated on the two algorithms. In terms of time complexity, FP-growth gives better performance.

Weblog mining is used on huge weblog file sources to discover automatically and analyze the wealth of useful emerging and user behavioral patterns. It is a tremendous task to identify the task of website users.

A. The top-down approach in frequent mining itemsets

The improved FP-growth algorithm is based on a top-down approach, i.e., TD-IFP-Growth is introduced [6]. Item name, count, node-link, and flag are four attributes of an improved FP-growth algorithm. Node link links the nodes with identical items, which helps to look for a specific node rapidly. Thus TD-IFP-Growth algorithm quickly traverses the tree. The FP-tree adopts a top-down approach in which conditional pattern base and sub-FP-trees are generated in the existing FP- Growth algorithm. At the same time, the proposed TD- IFP-Growth overcomes the problem of the existing FP-Growth algorithm by searching the IFP-tree, which is quite the opposite of the FP-Growth algorithm. The TD-IFP-Growth algorithm consumes less time and memory because it will not generate the conditional pattern.

B. Different approaches for frequent itemset mining

Data mining means retrieving hidden analytical information from huge databases; it is helping the organizations as they focus only on essential information in their data warehouses. Data mining tools are used for future development and performances, allowing organizations to create proactive ideas for decision-making systems. The traditional data mining problem is Frequent Itemset Mining, as it requires huge computations and input and output traffic capacity. One of the approaches which runs on the Hadoop cluster is one of the recent popular distributed frameworks which focus on parallel processing. The proposed framework extends the characteristics of the Apriori algorithm, which is related to the frequent itemset invention. The performance and scalability are highly improved when compared to the existing approaches. The algorithm which is proposed is tested on large data-sets distributed system on heterogeneous cluster [5]

.

IV.CONCLUSION

Data mining means retrieving hidden analytical information from huge databases; it is helping the organizations as they focus only on essential information in their data warehouses. Data mining

tools are used for future development and performances, allowing organizations to create proactive ideas for decision-making systems. Association Mining aims to extract correlations, frequent patterns, and association structures that are attention-grabbing among a 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 a minimum Support threshold and a minimum confidence threshold.

Web usage mining is the mining method used for user browsing and access patterns. At the side of the website, weblog mining is used to identify the web users to capture the data along with their browsing behaviors. This paper mainly aims to describe user behavior in classifying the patterns of the browsing and navigation data of web users and also measure the performance of the Frequent Pattern Growth algorithm and Apriori algorithm by comparing their performances.

REFERENCES

[1] https://www.singularities.com/blog/our-blog-1/post/apriori-vs-fp-growth-for-frequent-item-set-mining-11.
[2] M.S. Mythili, A.R. Mohamed Shanavas, Performance Evaluation of Apriori and FP-Growth Algorithms
[3] https://www.digitalvidya.com/blog/apriori-algorithms-in-data-mining/
[4] K.Dharmaraajan,M.A. Dorairangaswamy Analysis of FP-Growth and Apriori Algorithms on Pattern Discovery from Weblog Data
[5] Different Approaches for Frequent Itemset Mining P.V. Nikam1*, DS. Deshpande
[6] Improved FP-Growth Algorithm Based on Top-Down Approach in Mining Frequent Itemsets