Question: Implement the brute force method and compare

Find solutions for your homework   ◯

Implement the brute force method and compare the brute force method with the Apriori algorithm on each of the 5 transaction databases you created. Present computation (CPU or clock) time to demonstrate that the Apriori algorithm is faster than the brute force method on each of the 5 transaction databases. The brute force method and Apriori algorithm should output the same association rules on each database in any programming language (prefer python)

Answers

Anonymous
answered this

Brute force approach:-

A brute force approach is an approach that finds all the possible solutions to find a satisfactory solution to a given problem. The brute force algorithm tries out all the possibilities till a satisfactory solution is not found.

Such an algorithm can be of two types:

- **Optimizing:** In this case, the best solution is found. To find the best solution, it may either find all the possible solutions to find the best solution or if the value of the best solution is known, it stops finding when the best solution is found. For example: Finding the best path for the travelling salesman problem. Here best path means that travelling all the cities and the cost of travelling should be minimum.

- **Satisficing:** It stops finding the solution as soon as the satisfactory solution is found. Or example, finding the travelling salesman path which is within 10% of optimal.

- Often Brute force algorithms require exponential time. Various heuristics and optimization can be used:

- **Heuristic:** A rule of thumb that helps you to decide which possibilities we should look at first.

- **Optimization:** A certain possibilities are eliminated without exploring all of them.

Let's understand the brute force search through an example.

Suppose we have converted the problem in the form of the tree shown as below:
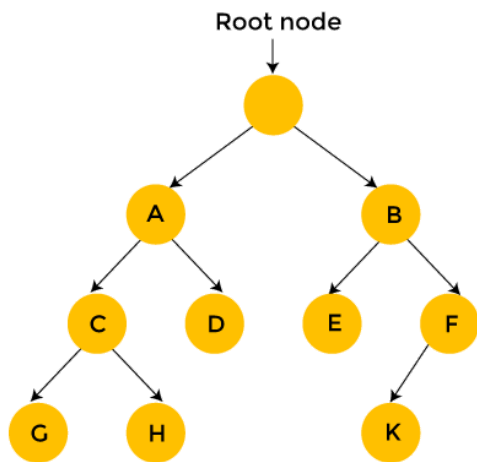
Brute force search considers each and every state of a tree, and the state is represented in the form of a node. As far as the starting position is concerned, we have two choices, i.e., A state and B state. We can either generate state A or state B. In the case of B state, we have two states, i.e., state E and F.

In the case of brute force search, each state is considered one by one. As we can observe in the above tree that the brute force search takes 12 steps to find the solution.

On the other hand, backtracking, which uses Depth-First search, considers the below states only when the state provides a feasible solution. Consider the above tree, start from the root node, then move to node A and then node C. If node C does not provide the feasible solution, then there is no point in considering the states G and H. We backtrack from node C to node A. Then, we move from node A to node D. Since node D does not provide the feasible solution, we discard this state and backtrack from node D to node A.

We move to node B, then we move from node B to node E. We move from node E to node K; Since k is a solution, so it takes 10 steps to find the solution. In this way, we eliminate a greater number of states in a single iteration. Therefore, we can say that backtracking is faster and more efficient than the brute force approach.

Advantages of a brute-force algorithm

The following are the advantages of the brute-force algorithm:

- This algorithm finds all the possible solutions, and it also guarantees that it finds the correct solution to a problem.

- This type of algorithm is applicable to a wide range of domains.

- It is mainly used for solving simpler and small problems.

- It can be considered a comparison benchmark to solve a simple problem and does not require any particular domain knowledge.

Apriori Algorithm:-

Apriori algorithm is an interesting approach to know what we need to purchase or tell the suggestions of our need. We all know that there is some kind of approach available on the e-commerce platform. It's none other than that, Amazon, Flipkart, Snapdeal, and etc. When we try to purchase an item in the e-shopping, the application will give us suggestions that we may buy together. It predicts other customers who frequently buy things together. This algorithm also allows us to know the prediction of things in multiple approaches.

"Apriori algorithm is an approach to identify the frequent itemset mining using association rule learning over the dataset and finds the trends over data."

This algorithm is widely used in market basket analysis and requires a larger amount of dataset. So, the approach can try sufficient combinations and occurrences of items to attain the result from each transaction.

Use of the Apriori Algorithm

Apriori algorithm works based on conditional rules, and it is considered as a classic algorithm among mining algorithms. Apriori helps to work efficiently by carrying out the mining association rules. Other traditional algorithms had a bottleneck in itemset generation and faced high consumption in time. The main use of this algorithm to mine the dataset by enhancing the user interest and identify the importance of itemsets and generate the frequent occurrences of an itemset. It follows certain approaches,

1. Handles and ready are the datasets

2. Applies mining association rule

- Identify frequent itemset and generates a set of data.

- Creates rules to find an efficient association.

3. Explore the interpretations using histograms, graphical representations.

Importance of Apriori Algorithm

- Increases the efficiency of search assumptions

- Enhances the performance of frequent set identification

- Transaction reduction is improvised – eliminates the less frequent sets in subsequent scans

- Includes hash-based counting.

- Eases the construction of user interests.

- Identifies the importance of different itemsets.

- The support function helps to identify different types of importance in itemsets.

- Storage space is reduced with the help of unnecessary itemset reduction.

- Improvised accuracy and efficiency of the algorithm.

- Works on supervised learning.

Different approaches in different languages

Apriori algorithm in data mining can be achieved in different languages like Python, R, etc. The main role of the algorithm is to find an association rule efficiently. And it is considered as the primary rule of the mining. The requisites of the association rules are,

- Finding the possible ways or rules holding its support value greater than its threshold support

- And its confidence values more than threshold confidence.

In Python, the papers have been accomplished in two possible ways. They are,

1. **Using the Brute force method –** *This is a longer process. First, rules are listed out and identify the support & confidence level on each rule. Then eliminates the value which is below its threshold support & confidence.*

2. **Using 2 – Step method –** *This process is much better than Brute force. The first step identifies the frequencies of items and forms a table. As a result, itemsets are found greater than threshold support. The second step uses binary partition on frequent sets and creates rules called candidate rules.*

In the R language, there are projects discussed in public forums. Some of the techniques are discussed here.

"Apriori's approach is an iterative approach, where it uses k-item set to search (k+1) itemsets. So the first itemset is found by gathering the count of each itemset. So it uses 1st itemset to find 2nd and goes on till no itemset can be explored.

An itemset is called a mathematical set of products in a basket."

Step #1 – Build the data and make it structural for data analysis. For Eg: We can take a comic book store as a case study.

Step #2 – The .csv file is used containing book details of the Comic book store. And the most interesting part is, we are using DC and Marvel collections for data mining.
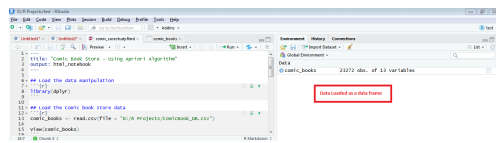
Step #3 – For the Apriori algorithm, R provides a package called **"arules"**. This package allows us to compute and inspect the algorithm's computation. To install and load the package from CRAN.
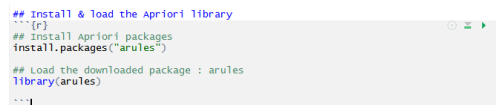
Step #4 – When we execute apriori's function, a class is created with the set of parameters. They are Support, Confidence, and Lift.
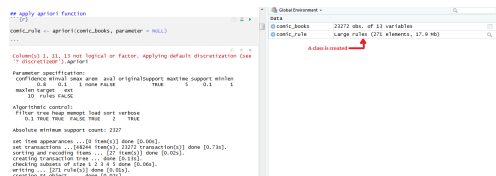
Here we can set the parameter as NULL or set with support = 0.001 as minimum value & confidence between 0.75 and 0.9. This change in support & confidence will lead to varied results.

Support: It is the basic probability of an event to occur. An event to get a product A, Support(A) is the no.of.transactions including A divided by total transactions.

Confidence: It is the conditional probability of the occurrence in the event. The change that happens in product A had already happened in product B.

Association Rules:-

Association Rule Mining is defined as:

"Let I= { …} be a set of 'n' binary attributes called items. Let D= { ….} be set of transaction called database. Each transaction in D has a unique transaction ID and contains a subset of the items in I. A rule is defined as an implication of form X->Y where X, Y? I and X?Y=?. The set of items X and Y are called antecedent and consequent of the rule respectively."

Learning of Association rules is used to find relationships between attributes in large databases. An association rule, A=> B, will be of the form" for a set of transactions, some value of itemset A determines the values of itemset B under the condition in which minimum support and confidence are met".

Support and Confidence can be represented by the following example:

Bread=> butter [support=2%, confidence-60%]

The above statement is an example of an association rule. This means that there is a 2% transaction that bought bread and butter together and there are 60% of customers who bought bread as well as butter.

Support and Confidence for Itemset A and B are represented by formulas:

$$\text{Support (A)} = \frac{\text{Number of transaction in which A appears}}{\text{Total number of transactions}}$$

$$\text{Confidence (A} \rightarrow \text{B)} = \frac{\text{Support(AUB)}}{\text{Support(A)}}$$

Association rule mining consists of 2 steps:

1. Find all the frequent itemsets.

2. Generate association rules from the above frequent itemsets.

Why Frequent Itemset Mining?

Frequent itemset or pattern mining is broadly used because of its wide applications in mining association rules, correlations and graph patterns constraint that is based on frequent patterns, sequential patterns, and many other data mining tasks.

Apriori Algorithm – **Frequent Pattern Algorithms**

Apriori algorithm was the first algorithm that was proposed for frequent itemset mining. It was later improved by R Agarwal and R Srikant and came to be known as Apriori. This algorithm uses two steps "join" and "prune" to reduce the search space. It is an iterative approach to discover the most frequent itemsets.

Apriori says:

The probability that item I is not frequent is if:

- P(I) < minimum support threshold, then I is not frequent.

- P (I+A) < minimum support threshold, then I+A is not frequent, where A also belongs to itemset.

- If an itemset set has value less than minimum support then all of its supersets will also fall below min support, and thus can be ignored. This property is called the Antimonotone property.

The steps followed in the Apriori Algorithm of data mining are:

1. **Join Step**: This step generates (K+1) itemset from K-itemsets by joining each item with itself.

2. **Prune Step**: This step scans the count of each item in the database. If the candidate item does not meet minimum support, then it is regarded as infrequent and thus it is removed. This step is performed to reduce the size of the candidate itemsets.

Steps In Apriori

Apriori algorithm is a sequence of steps to be followed to find the most frequent itemset in the given database. This data mining technique follows the join and the prune steps iteratively until the most frequent itemset is achieved. A minimum support threshold is given in the problem or it is assumed by the user.

#1) In the first iteration of the algorithm, each item is taken as a 1-itemsets candidate. The algorithm will count the occurrences of each item.

#2) Let there be some minimum support, min_sup (eg 2). The set of 1 – itemsets whose occurrence is satisfying the min sup are determined. Only those candidates which count more than or equal to min_sup, are taken ahead for the next iteration and the others are pruned.

#3) Next, 2-itemset frequent items with min_sup are discovered. For this in the join step, the 2-itemset is generated by forming a group of 2 by combining items with itself.

#4) The 2-itemset candidates are pruned using min-sup threshold value. Now the table will have 2 – itemsets with min-sup only.

#5) The next iteration will form 3 –itemsets using join and prune step. This iteration will follow antimonotone property where the subsets of 3-itemsets, that is the 2 –itemset subsets of each group fall in min_sup. If all 2-itemset subsets are frequent then the superset will be frequent otherwise it is pruned.

#6) Next step will follow making 4-itemset by joining 3-itemset with itself and pruning if its subset does not meet the min_sup criteria. The algorithm is stopped when the most frequent itemset is achieved.
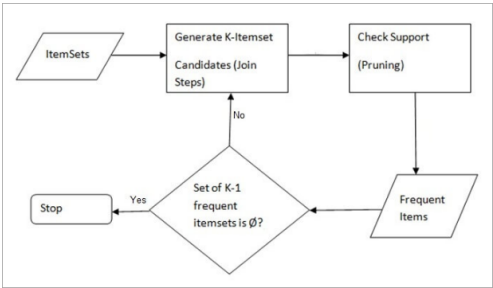
0 Comments

Was this answer helpful?   👍 0   👎 0