Apriori Algorithm

Apriori algorithm refers to the algorithm which is used to calculate the association rules between objects. It means how two or more objects are related to one another. In other words, we can say that the apriori algorithm is an association rule leaning that analyzes that people who bought product A also bought product B.

The primary objective of the apriori algorithm is to create the association rule between different objects. The association rule describes how two or more objects are related to one another. Apriori algorithm is also called frequent pattern mining. Generally, you operate the Apriori algorithm on a database that consists of a huge number of transactions. Let's understand the apriori algorithm with the help of an example; suppose you go to Spencer and buy different products. It helps the customers buy their products with ease and increases the sales performance of Spencer.

Introduction

We take an example to understand the concept better. You must have noticed that the Pizza shop seller makes a pizza, soft drink, and breadstick combo together. He also offers a discount to their customers who buy these combos. Do you ever think why does he do so? He thinks that customers who buy pizza also buy soft drinks and breadsticks. However, by making combos, he makes it easy for the customers. At the same time, he also increases his sales performance.

Similarly, you go to Spencer, and you will find biscuits, chips, and Chocolate bundled together. It shows that the shopkeeper makes it comfortable for the customers to buy these products in the same place.

What is Apriori Algorithm?

Apriori algorithm refers to an algorithm that is used in mining frequent products sets and relevant association rules. Generally, the apriori algorithm operates on a database containing a huge number of transactions. For example, the items customers but at a Big Bazar.

Apriori algorithm helps the customers to buy their products with ease and increases the sales performance of the particular store.

Components of Apriori algorithm

The given three components comprise the apriori algorithm.

1. Support
2. Confidence
3. Lift

Example

We have already discussed above; you need a huge database containing a large no of transactions. Suppose you have 4000 customers transactions in Walmart. You have to calculate the Support, Confidence, and Lift for two products, and you may say Biscuits and Chocolate. This is because customers frequently buy these two items together.

Out of 4000 transactions, 400 contain Biscuits, whereas 600 contain Chocolate, and these 600 transactions include a 200 that includes Biscuits and chocolates. Using this data, we will find out the support, confidence, and lift of Chocolate.

Support

Support refers to the default popularity of any product. You find the support as a quotient of the division of the number of transactions comprising that product by the total number of transactions. Hence, we get

$$supp(X \Rightarrow Y) = \frac{|X \cup Y|}{n}$$

Support (Biscuits) = (Transactions relating biscuits) / (Total transactions)

= 400/4000 = 10 percent.

Confidence

Confidence refers to the possibility that the customers bought both biscuits and chocolates together. So, you need to divide the number of transactions that comprise both biscuits and chocolates by the total number of transactions to get the confidence.

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}$$

Hence,

Confidence = (Transactions relating both biscuits and Chocolate) / (Total transactions involving Biscuits)

= 200/400

= 50 percent.

It means that 50 percent of customers who bought biscuits bought chocolates also.

Lift

Consider the above example; lift refers to the increase in the ratio of the sale of chocolates when you sell biscuits. The mathematical equations of lift are given below.

$$lift(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)supp(Y)}$$

Lift = (Confidence (Biscuits - chocolates)/ (Support (Biscuits))

= 50/10 = 5

It means that the probability of people buying both biscuits and chocolates together is five times more than that of purchasing the biscuits alone. If the lift value is below one, it requires that the people are unlikely to buy both the items together. Larger the value, the better is the combination.

Conviction

Consider the above example;

$$conv(X \Rightarrow Y) = \frac{1 - supp(Y)}{1 - conf(X \Rightarrow Y)}$$

Conviction=(1-Support(Biscuits))/(1-confidence(Biscuits,Chocolates))

=1-0.10/1-0.5

=0.9/0.5=1.8

How does the Apriori Algorithm work in Data Mining?

Transaction ID	Rice	Pulse	Oil	Milk	Apple
t1	1	1	1	0	0
t2	0	1	1	1	0
t3	0	0	0	1	1
t4	1	1	0	1	0
t5	1	1	1	0	1
t6	1	1	1	1	1

We will understand this algorithm with the help of an example

Consider a Big Bazar scenario where the product set is P = {Rice, Pulse, Oil, Milk, Apple}. The database comprises six transactions where 1 represents the presence of the product and 0 represents the absence of the product.

The Apriori Algorithm makes the given assumptions

- All subsets of a frequent itemset must be frequent.
- The subsets of an infrequent item set must be infrequent.
- Fix a threshold support level. In our case, we have fixed it at 50 percent.

Step 1

Make a frequency table of all the products that appear in all the transactions. Now, short the frequency table to add only those products with a threshold support level of over 50 percent. We find the given frequency table.

Product	Frequency (Number of transactions)
Rice (R)	4
Pulse(P)	5
Oil(O)	4
Milk(M)	4

The above table indicated the products frequently bought by the customers.

Step 2

Itemset	Frequency (Number of transactions)
Rice, Pulses	4
Rice, Oil	3
Rice, Milk	2
Pulses, Oil	4
Pulses, Milk	3
Oil, Milk	2

Create pairs of products such as Rice Pulses, Rice Oil, Rice Milk, Pulses Oil, Pulses Milk, Oil Milk. You will get the given frequency table.

Step 3

Implementing the same threshold support of 50 percent and consider the products that are more than 50 percent. In our case, it is more than 3

Thus, we get Rice, Pulses; Rice, Oil; Pulses, Oil; and Pulses, Milk.

Step 4

Now, look for a set of three products that the customers buy together. We get the given combination.

1. Rice,Pulses and Rice,Oil give Rice,Pulses,Oil
2. Pulses,Oil and Pulses,Milk give Pulses,Oil,Milk

Step 5

Calculate the frequency of the two itemsets, and you will get the given frequency table.

Itemset	Frequency (Number of transactions)
Rice,Pulses,Oil	4
Pulses,Oil,Milk	3

If you implement the threshold assumption, you can figure out that the customers' set of three products is Rice,Pulses,Oil

We have considered an easy example to discuss the apriori algorithm in data mining. In reality, you find thousands of such combinations.

How to improve the efficiency of the Apriori Algorithm?

There are various methods used for the efficiency of the Apriori algorithm

Hash-based itemset counting

In hash-based itemset counting, you need to exclude the k-itemset whose equivalent hashing bucket count is least than the threshold is an infrequent itemset.

Transaction Reduction

In transaction reduction, a transaction not involving any frequent X itemset becomes not valuable in subsequent scans.

Apriori Algorithm in data mining

We have already discussed an example of the apriori algorithm related to the frequent itemset generation. Apriori algorithm has many applications in data mining.

The primary requirements to find the association rules in data mining are given below.

Use Brute Force

Analyze all the rules and find the support and confidence levels for the individual rule. Afterward, eliminate the values which are less than the threshold support and confidence levels.

The two-step approaches

The two-step approach is a better option to find the associations rules than the Brute Force method.

Step 1

In this article, we have already discussed how to create the frequency table and calculate itemsets having a greater support value than that of the threshold support.

Step 2

To create association rules, you need to use a binary partition of the frequent itemsets. You need to choose the ones having the highest confidence levels.

In the above example, you can see that the Rice-Pulses-Oil combination was the frequent itemset. Now, we find out all the rules using Rice-Pulses-Oil.

RP-O, RO-P, PO-R, O-RP, P-RO, R-PO(R=Rice,P=Pulses,O=Oil)

You can see that there are six different combinations. Therefore, if you have n elements, there will be 2^n - 2 candidate association rules.

Advantages of Apriori Algorithm

- It is used to calculate large itemsets.
- Simple to understand and apply.

Disadvantages of Apriori Algorithms

- Apriori algorithm is an expensive method to find support since the calculation has to pass through the whole database.
- Sometimes, you need a huge number of candidate rules, so it becomes computationally more expensive.

Limitations of Apriori Algorithm

Apriori Algorithm can be slow. The main limitation is time required to hold a vast number of candidate sets with much frequent itemsets, low minimum support or large itemsets i.e. it is not an efficient approach for large number of datasets. For example, if there are 10^4 from frequent 1- itemsets, it need to generate more than 10^7 candidates into 2-length which in turn they will be tested and accumulate. Furthermore, to detect frequent pattern in size 100 i.e. v1, v2… v100, it have to generate 2^100 candidate itemsets that yield on costly and wasting of time of candidate generation. So, it will check for many sets from candidate itemsets, also it will scan database many times repeatedly for finding candidate itemsets. Apriori will be very low and inefficiency when memory capacity is limited with large number of transactions.