Govardhan Seshasayee

Association Rule Mining: Market Basket Analysis

Overview:

Association rule mining is a technique to identify underlying relations between different items. Take an example of a Super Market where customers can buy variety of items. Usually, there is a pattern in what the customers buy. For instance, mothers with babies buy baby products such as milk and diapers. Damsels may buy makeup items whereas bachelors may buy beers and chips etc. In short, transactions involve a pattern. More profit can be generated if the relationship between the items purchased in different transactions can be identified.

For instance, if item A and B are bought together more frequently then several steps can be taken to increase the profit. For example:

A and B can be placed together so that when a customer buys one of the product he doesn't have to go far away to buy the other product.

People who buy one of the products can be targeted through an advertisement campaign to buy the other.

Collective discounts can be offered on these products if the customer buys both of them.

Both A and B can be packaged together.

The process of identifying an associations between products is called association rule mining.

Association rule learning is the rule-based machine learning method for discovering interesting relations between variables in large databases using some measure of interestingness. Apriori algorithm is one such algorithm that is used to identify these strong rules. It is an algorithm for frequent item set mining and association rule learning over relational databases.

Introduction:

Frequent pattern mining algorithm is one of the most important techniques of data mining to discover relationships between different items in a dataset. These relationships are represented in the form of association rules. Apriori is an algorithm used to identify frequent item sets (in our case, item pairs). It does so by using a "bottom up" approach, first identifying individual items that satisfy a minimum occurrence threshold. It then extends the item set, adding one item at a time and checking if the resulting item set still satisfies the specified threshold. The algorithm stops when there are no more items to add that meet the minimum occurrence requirement. A set of items together is called an Itemset. An itemset that occurs frequently is called frequent itemset. A set of items is called frequent if it satisfies a minimum threshold value for support & confidence.

Association rule mining is defined as:

“Let I = {….} be a set of ‘n’ binary attributes called items. Let D = {….} be the set of transactions called database. Each transaction in D has a unique transaction ID and contains a subset of item in I. A rule is defined as an implication of the form A->B where A, B (subset

symbol) I. The set of items A and B are called antecedent and consequent of the rules respectively.”

Various Metric in Measure Association:

There are five key metrics to consider when evaluating association rules-

Support

This is the percentage of orders that contain the item set. The minimum support required by apriori can be set based on knowledge of your domain. In the grocery dataset for example, since there could be thousands of distinct items and an order can contain only a small fraction of these items, setting the support threshold to 0.01% may be reasonable.

Confidence

Given two items, A and B, confidence measures the percentage of items that B is purchased, given that item A was purchased. This is expressed as:

Confidence(A->B) = support (A, B) / support (A)

Confidence values range from 0 to 1, where 0 indicates that B is never purchased when A is purchased, and 1 indicates that B is always purchased whenever A is purchased. Note that the confidence measure is directional. This means that we can also compute the percentage of times that items A is purchased, given that item B was purchased.

Confidence(B->A) = support (A, B) / support(B)

A confidence value of 0.75 implies that out of all orders that contain A, 75% of them also contain B.

Lift

Given two items, A and B, lift indicates whether there is a relationship between A and B, or whether the two items are occurring together in the same orders simply by chance (i.e. at random). Unlike the confidence metric whose value may vary depending on the direction, lift has no direction. This means that the lift (A, B) is always equal to the lift (B, A).

Lift (A, B) = Lift (B, A) = Confidence(B->A)/support(A) = support (A, B)/(support(A) * support(B))

One way to understand lift is to think of the denominator as the likelihood that A and B will appear in the same order if there was no relationship between them. If suppose A occurred in 80% of the orders and B occurred in 60% of the orders, then if there was no relationship between them, we would expect both of them to show up together in the same order 48% of the time (ie: 80% * 60%). The numerator, on the other hand, represents how often A and B actually appear together in the same order. Taking the numerator and dividing it by the denominator, we get to know how many more times A and B actually appear in the same order, compared to if there was no relationship between them (i.e.: that they are occurring together simply at random).

In summary, lift can take on the following values: - Lift = 1 implies no relationship between A and B (ie: A & B occur together only by chance). - Lift > 1 implies that there is a positive

relationship between A & B. (i.e.: A & B occur together more often than random). - Lift < 1 implies that there is a negative relationship between A & B (i.e.: A & B occur together less often than random).

Leverage

levarage($A \rightarrow C$)=support($A \rightarrow C$)−support(A)×support(C),range: [−1,1]levarage(A→C)=support(A→C)−support(A)×support(C),range: [−1,1]

Leverage computes the difference between the observed frequency of A and C appearing together and the frequency that would be expected if A and C were independent. An leverage value of 0 indicates independence.

Conviction

conviction($A \rightarrow C$)=1−support(C)1−confidence($A \rightarrow C$),range: [0,∞]conviction(A→C)=1−support(C)1−confidence(A→C),range: [0,∞]

A high conviction value means that the consequent is highly depending on the antecedent. For instance, in the case of a perfect confidence score, the denominator becomes 0 (due to 1 - 1) for which the conviction score is defined as 'inf'. Similar to lift, if items are independent, the conviction is 1.

Steps involved in Apriori Algorithm

For large sets of data, there can be hundreds of items in hundreds of thousands transactions. The Apriori algorithm tries to extract rules for each possible combination of items. For instance, Lift can be calculated for item 1 and item 2, item 1 and item 3, item 1 and item 4 and then item 2 and item 3, item 2 and item 4 and then combinations of items e.g. item 1, item 2 and item 3; similarly item 1, item2, and item 4, and so on.

As you can see from the above example, this process can be extremely slow due to the number of combinations. To speed up the process, we need to perform the following steps:

Set a minimum value for support and confidence. This means that we are only interested in finding rules for the items that have certain default existence (e.g. support) and have a minimum value for co-occurrence with other items (e.g. confidence).

Extract all the subsets having higher value of support than minimum threshold.

Select all the rules from the subsets with confidence value higher than minimum threshold.

Order the rules by descending order of Lift.