**LAB-9 Association Rule Learning**

**TITLE:**

**Theory:**

**Apriori** algorithm works on the principle of Association Rule Mining.  
association rule mining is a technique to identify underlying relations between different items. This relationship can be a similarity between items on how frequently they are bought or how similar users bought it. In this article, we will be looking on how the Apriori algorithm works with a python example. In the supermarket, the Apriori algorithm can be used to keep similar items together. like shaving foam, shaving cream and other men's grooming products can be kept adjacent to each other based on the order or popularity that they are bought together. So that it becomes easy for customers to buy the product and can add in more business and profit to the supermarket. Now that we have a basic idea of Apriori algo, now we will into the theory of Apriori algo.

**Theory of Apriori Algorithm.**

The Apriori algorithm was proposed by Agrawal and Srikant in 1994.

There are three major components of the Apriori algorithm:  
1) **Support**  
2) **Confidence**  
3) **Lift**

Suppose we have a record of 1000 customers transactions and we want to find out support, confidence and lift for milk and diapers. out of 1000 transactions, 120 contains a milk and 150 contains a diaper. out of these 150 transactions where a diaper is purchased 30 contains transaction contains milk as well. we will use this data to calculate support, confidence and lift.

**Support**support refers to the popularity of item and can be calculated by finding the number of transactions containing a particular item divided by the total number of transactions.

Support(diaper) = (Transactions containing (diaper))/(Total Transactions)  
Support(diaper) = 150 / 1000 = 15 %

**Confidence**

Confidence refers to the likelihood that an item B is also bought if item A is bought. It can be calculated by finding the number of transactions where A and B are bought together, divided by the total number of transactions where A is bought. Mathematically, it can be represented as:

Confidence (A → B) = (Transactions containing both (A and B))/(Transactions containing A)

The confidence of likelihood of purchasing a diaper if a customer purchase milk.  
Confidence (milk → diaper) = (Transactions containing both (milk and diaper))/(Transactions containing milk)  
Confidence(milk → diaper) =30 / 120 = 25 %

Confidence is similar to Naive Based Algorithm.

**Lift**

Lift refers to the increase in the ratio of the sale of B when A is sold.   
Lift (A –> B) can be calculated by dividing Confidence (A -> B) divided by Support(B).   
Mathematically it can be represented as:  
Lift(A→B) = (Confidence (A→B))/ (Support (B))

Lift (milk → diaper) = (Confidence (milk → diaper))/ (Support (diaper))  
Lift (milk → diaper) = 25 / 15 = 1.66

So, by Lift theory, there is 1.66 times more chance of buying milk and diaper together then just buying diaper alone.

**Association rule by Lift**  
lift = 1 → There is no association between A and B.  
lift < 1→ A and B are unlikely to be bought together.  
lift > 1 → greater the lift greater is the likelihood of buying both products together.

**ALGORITHM:**

1. 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).
2. Extract all the subsets having a higher value of support than a minimum threshold.
3. Select all the rules from the subsets with confidence value higher than the minimum threshold.
4. Order the rules by descending order of Lift.

**CODE: -**

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 28 10:24:12 2019

@author: RoKu

"""

# Apriori

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from apyori import apriori

# Data Preprocessing

store\_data = pd.read\_csv('store\_data.csv',header=None)

transactions = []

for i in range(0, 7501):

transactions.append([str(dataset.values[i,j]) for j in range(0, 20)])

# Training Apriori on the dataset

rules = apriori(transactions, min\_support = 0.0045, min\_confidence = 0.2, min\_lift = 3, min\_length = 2)

# Visualising the results

results = list(rules)

#print

print(len(list(rules))

**RESULT:**



**CONFIDENCES**

|  |  |
| --- | --- |
| **SUBMISSION DATE**-  **28-03-19** | **SIGN OF COURSE INSTRUCTOR**- |
| **ROLL NO OF THE STUDENT-**  **01** | **NAME OF THE STUDENT-**  **Rohit Kulkarni** |