Shop Conveniently with Apriori Algorithm

Robin Bista

03/20/2021

Abstract

Apriori algorithm is commonly used to extract frequently purchased items from a large data set and create a correlation of the items. Companies of all flavors, big or small, have ever-growing data to be sorted, analyzed, and archived as per the company's compliance. Apriori algorithm identifies association between customers regular choice of items and provides in-advance suggestions for timely refilling of the inventory.

Background

My Walmart experience of ending up buying more items than my actual shopping list should be mutual to my readers. In today's world retailers think strategically and plan ahead of their customers' needs to optimize sales. It is a competitive market where customers' needs move with a high velocity and negligence in predicting customer choice and demand hampers the business and eventually diminishes the customer base. Let us explore the Apriori algorithm to verify whether retailers can successfully predict customers' demand and place items in adjacent aisles to influence buyers to purchase more than their initial needs.

Objective

The objective of this proposition is to leverage the Apriori algorithm to group associated items in a set and station them in the same aisle for a convenient shopping experience for customers and sales optimizations for the retailers. Let us explore the incorporated Grocery data in R and construct rules to group items in a set that are frequently purchased. The correlation of items will be evaluated on the basis of lift, support, and confidence.

Dataset Analysis

Before we begin applying the "Apriori" algorithm to our dataset, we need to extract and transform the dataset to a type "Transactions".

```
## Formal class 'transactions' [package "arules"] with 3 slots
##   ..@ data       :Formal class 'ngCMatrix' [package "Matrix"] with 5 slots
##   .. .. ..@ i       : int [1:43367] 13 60 69 78 14 29 98 24 15 29 ...
##   .. .. ..@ p       : int [1:9836] 0 4 7 8 12 16 21 22 27 28 ...
##   .. .. ..@ Dim     : int [1:2] 169 9835
##   .. .. ..@ Dimnames:List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : NULL
##   .. .. ..@ factors : list()
##   ..@ itemInfo   :'data.frame':  169 obs. of  3 variables:
##   .. ..$ labels: chr [1:169] "frankfurter" "sausage" "liver loaf" "ham" ...
##   .. ..$ level2: Factor w/ 55 levels "baby food","bags",..: 44 44 44 44 44 44 44 42 42 4
##   .. ..$ level1: Factor w/ 10 levels "canned food",..: 6 6 6 6 6 6 6 6 6 6 ...
##   ..@ itemsetInfo:'data.frame':  0 obs. of  0 variables
```

Figure 1: Transaction dataset

The structure of our transaction, dataset shows that it is internally divided into three slots: Data, itemInfo, and itemsetInfo. The “Data” slot contains dimensions, dimension names, and other numerical values which represent the number of products sold in each transaction.

```
## transactions as itemMatrix in sparse format with
##  9835 rows (elements/itemsets/transactions) and
##  169 columns (items) and a density of 0.02609146
##
## most frequent items:
##       whole milk other vegetables       rolls/buns             soda
##             2513             1903             1809             1715
##           yogurt          (Other)
##             1372            34055
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77   55   46
##   17   18   19   20   21   22   23   24   26   27   28   29   32
##   29   14   14    9   11    4    6    1    1    1    1    3    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   4.409   6.000  32.000
##
## includes extended item information - examples:
##        labels  level2           level1
## 1 frankfurter sausage meat and sausage
## 2     sausage sausage meat and sausage
## 3  liver loaf sausage meat and sausage
```

Figure 2: Frequently purchased items (top 5)

The summary statistics show the top 5 items sold in a transaction set. For example, “Whole Milk”,” Other Vegetables”,” Rolls/Buns”,” Soda”, and “Yogurt”.

Applying the Algorithm

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 9
```

Figure 3: Parameter specification

The minimum support parameter (minSup) is set to .001 to include as many items as possible. We can set minimum confidence (minConf) to anywhere between 0.75 and 0.85 for varied results.

```
##      lhs                                rhs                support confidence    coverage
## [1]  {liquor,
##       red/blush wine}                => {bottled beer}     0.001931876  0.9047619 0.002135231 1
## [2]  {curd,
##       cereals}                       => {whole milk}       0.001016777  0.9090909 0.001118454  
## [3]  {yogurt,
##       cereals}                       => {whole milk}       0.001728521  0.8095238 0.002135231  
## [4]  {butter,
##       jam}                           => {whole milk}       0.001016777  0.8333333 0.001220132  
## [5]  {soups,
##       bottled beer}                  => {whole milk}       0.001118454  0.9166667 0.001220132  
## [6]  {napkins,
##       house keeping products}        => {whole milk}       0.001321810  0.8125000 0.001626843  
## [7]  {whipped/sour cream,
##       house keeping products}        => {whole milk}       0.001220132  0.9230769 0.001321810  
## [8]  {pastry,
##       sweet spreads}                 => {whole milk}       0.001016777  0.9090909 0.001118454  
## [9]  {turkey,
##       curd}                          => {other vegetables} 0.001220132  0.8000000 0.001525165  4
## [10] {rice,
##       sugar}                         => {whole milk}       0.001220132  1.0000000 0.001220132  
```

Figure 4: List of rules

The top 10 rules are derived from our Groceries dataset with the above code. The first rule shows if Liquor and Red Wine are bought, it's highly likely a bottled beer will also be purchased in that transaction.

Visualizing the Results

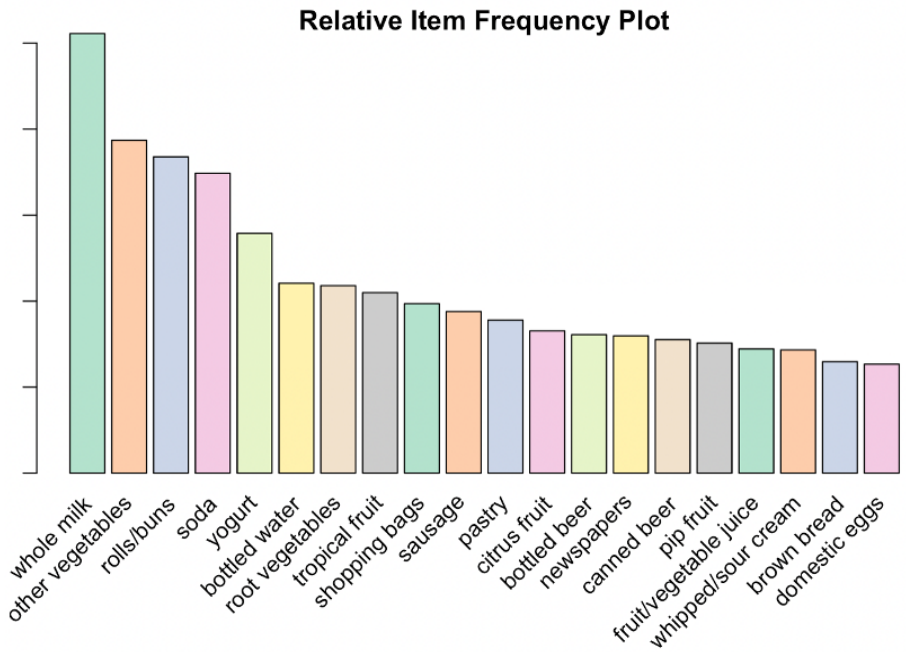

Figure 5: Relative item frequency

1. These histograms depict the number of times an item has occurred in the dataset compared to the others. The relative frequency plot accounts for the fact that “Whole Milk” and “Other Vegetables” constitute around half of the transaction dataset; half the sales of the store are these items.

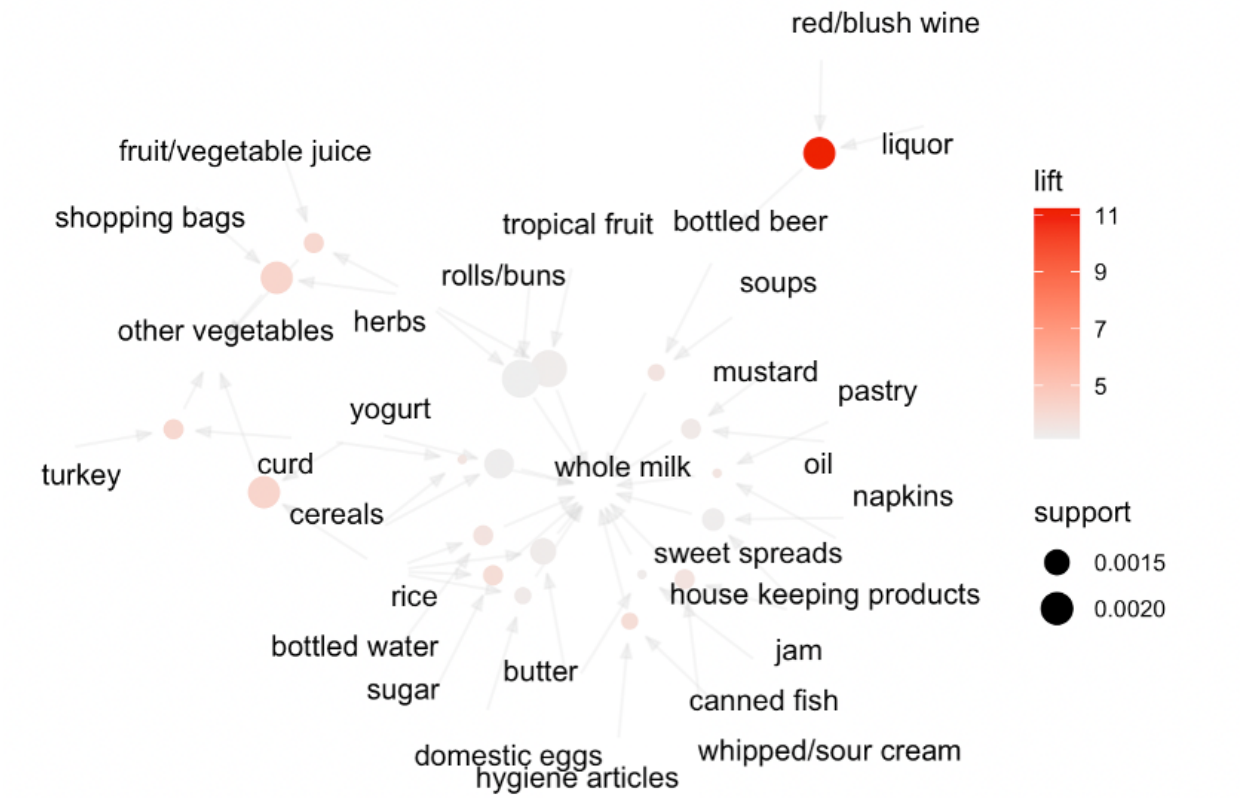

Figure 6: Graph Visualization

2. The above graph shows that most of the transactions were around “Whole Milk”. All liquor and wine are very strongly associated so those must be placed together. Another association is people buying tropical fruits and herbs, also buy rolls and buns. These items can be placed in an aisle together.

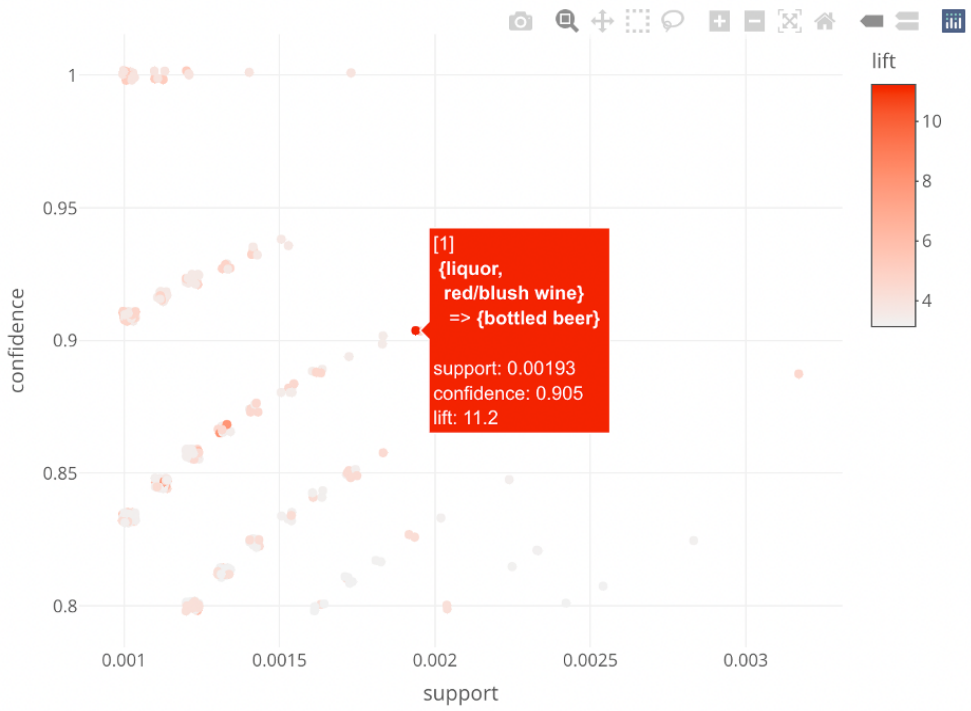

Figure 7: Interactive plot

3. The interactive scattered plot represents the rules which optimizes the sale of the store. Rules with higher lift and confidence are selected to evaluate the set of items effectively.

Finding and Explanation

The above interactive scatter plot shows rules with optimal lift, support, and confidence in general. It is clear that liquor, red/blush wine is related to bottle beer with lift = 11.2, support = 0.001, and confidence = 90%,. Similarly, tropical fruits, other vegetables, whole milk, and oil have a lift with a strong correlation with root vegetables with a lift = 7.95, support = 0.001, and confidence = 86%.

Also, Citrus Fruit, Grape, and Fruit/Vegetable Juice have a decent correlation with tropical fruit with lift = 8.06, support= 0.001, and confidence = 84%. Finally, we can also see that pastry, curd, cereals, and sweet spreads have high relation with milk, so it is best to place them in the same aisle.

Aisles Proposed

Liquor Aisle – Liquor, Red/Blush Wine, Bottled Beer

Groceries Aisle – Other vegetables, Whole milk, Oil, Yogurt, Rice, Root Vegetable

Fruit Aisle – Citrus Fruit, Grape, Fruit/Vegetable juice, Tropical fruits

Breakfast Aisle – Pastry, Curd, Cereals, Sweet Spreads

Conclusion

It demonstrates that the Apriori algorithm can be used to verify the relationship between products and help plan a convenient customer shopping experience along with sales boost of the retail store by beforehand knowledge of the customers' needs and shopping habits.

References

"Visualizing association rules" *Springer Link*, 07 May. 2016, Visualizing Rules

"R Markdown theme Gallery" *Andrew*, May. 2021, R Markdown

"Introduction to Association Rule Mining in R ", 14 May. 2021, Mining in R with Association rule

"Study on Apriori Algorithm and its Application in Grocery Store, 14 July. 2013, Study on Apriori

"Grocery Shopping Impulse Purchases with Apriori Algorithm and Association Rules in R" *RALGO*, 8 Oct. 2018, Grocery Shopping with Apriori

"Apriori Algorithm Explained" *Youtube*, 19 June. 2019, Apriori Algorithm Explained

"Product Recommendation Case Study Using Apriori Algorithm for a Grocery Store" *Medium*, 15 Jan. 2015, Shopping with Apriori