Association Rule Mining for a UK Online Retail Company

Alex Fung   Viswesh Krishnamurthy   Tony Lee   Patrick Osborne

Feb 26, 2020

Abstract

The retail industry is one that has changed far beyond recognition with the advent of internet. From once having to make a list, physically travel to a store, buy & haul your purchase yourself, to now simply ordering your needs online and getting it delivered in less than 24 hours, retail buying has become far more convenient. This buying convenience has also introduced challeges on the part of the sellers. The once obvious buying patterns are no longer obvious and requires complex analyses to understand customer preferences. In this project, we attempt to help a UK based Online Retail store understand their customers' buying patterns.

Background

One of the most powerful tools in online retail is a recommender system. Such a system helps sellers mine through their sales and unearth important associations between their products. In turn, such associations can be presented to customers as recommendations. Our client, the online retail store wishes to build a long term strategy based on the understanding this project gives them.

Objective

The objective of our analysis is to develop an usupervised, prediction model using Machine Learning techniques and the CRISP-DM framework (cite textbook) on the available transaction data to optimally place product that are frequently purchased together, or to suggest other items to purchase when certain items are added to the online shopping cart. The intent of this is to increase sales at this retailer by making it more convenient for the purchaser to find products related to the ones they intend to purchase and to upsell at the cashier or online checkout

Data Analysis

The original data set, "Online Retail.csv" is sourced from the UCI Machine Learning Repository. It is a transnational data set which contains all the transactions occurring between 01\12\2010 and 09\12\2011. The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers.

Data dictionary

Table 1: Data Dictionary - Online retail store

Feature	Feature.Description
InvoiceNo	Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
StockCode	Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
Description	Product (item) name. Nominal.
Quantity	The quantities of each product (item) per transaction. Numeric.
InvoiceDate	Invice Date and time. Numeric, the day and time when each transaction was generated.
UnitPrice	Unit price. Numeric, Product price per unit in sterling.
CustomerID	Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
Country	Country name. Nominal, the name of the country where each customer resides.

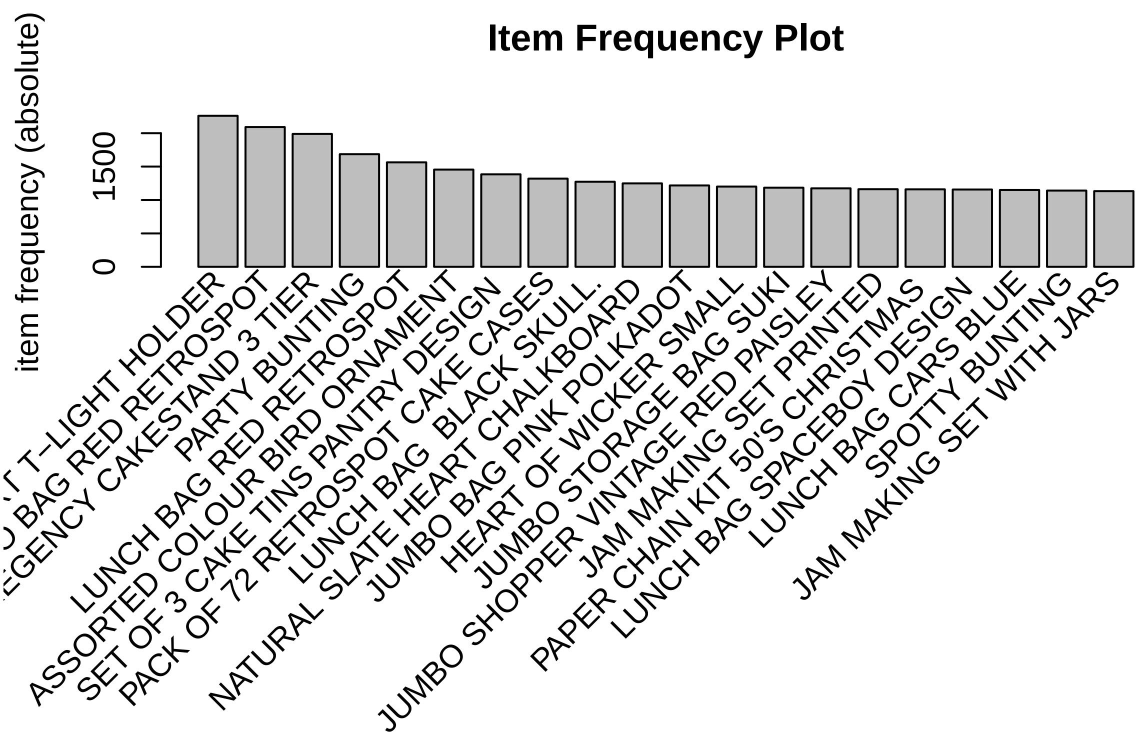
Initial Data Exploration & Cleaning

The original data set in this case is a rather simple data, with features that are obvious & straight forward. A quick look at the header of the data set, helps one understand this.

X	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/10 8:26	2.55	17850	United Kingdom
2	536365	71053	WHITE METAL LANTERN	6	12/1/10 8:26	3.39	17850	United Kingdom
3	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/10 8:26	2.75	17850	United Kingdom
4	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/10 8:26	3.39	17850	United Kingdom
5	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/10 8:26	3.39	17850	United Kingdom
6	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12/1/10 8:26	7.65	17850	United Kingdom

We learn from the original data that all invoices that start with a "C" are cancelled orders and therefore are removed from the data. Also, the patterns will be interpreted based on the product descriptions and hence all rows with empty descriptions are deleted too.

DOTCOM POSTAGE: Lines with description "DOTCOM POSTAGE" refer to postage charges and don't contribute anything meaningful to the problem. Hence those lines are removed as well. This now leaves us with a dataset that has all valid invoices and items



Models

From the onset, it was clear that an association rule mining model will answer the question. In this light, it was decided to use the following models for association rule mining

- Apriori algorithm
- FP Growth algorithm
- ECLAT algorithm

All association rule mining algorithms have 3 very important parameters

- Support
- Confidence
- Lift

Support:
Support is the proportion that an item represents in the total transaction dataset. Example, *support(PINK REGENCY TEACUP AND SAUCER) => (Trasanctions featuring (PINK REGENCY TEACUP AND SAUCER))/Total Transactions

Confidence:
Confidence is defined as the probability that an item combination was bought. Example, *confidence(PINK REGENCY TEACUP AND SAUCER & GREEN REGENCY TEACUP AND SAUCER) => (Total transactions with both PINK & GREEN TEACUP & SAUCER)/Total Transactions

Lift:
Lift is defined as the increase in the chance that an item combination is bought when a single item in that combination is bought. Example, *Lift(PINK & GREEN REGENCY TEACUP & SAUCER) = Confidence(PINK & GREEN REGENCY TEACUP & SAUCER)/support(GREEN REGENCY TEACUP & SAUCER)

Apriori algorithm

The apriori algorithm is one that is custom built for mining for association rules in a dataset. This algorithm works on datasets that are transactions. In our case, we have already converted the dataset into "transactions" prior to running the model on. The apriori algorithm, as the name suggests works on the basis of previously mined information. It is a breadth first algorithm, where it mines for frequent subsets by traversing across the breadth of each level of transaction.It starts by creating a "candidate set" which is a table of count of each unique item in the dataset. In the next step, it expands by counting each frequently appearing pair and then three items in the subsequent step and so on and so forth. At each step and finally, the stop criterion for the algorithm is decided by the "support" metric explained above

```
#apriori algorithm
apriori.rules <- apriori(trans, parameter = list(supp=0.01, conf=0.8,minlen = 3, maxlen=5))
apriori.rules_df<- data.frame(lhs = labels(lhs(apriori.rules)), rhs = labels(rhs(apriori.rules)),
                              apriori.rules@quality)
```

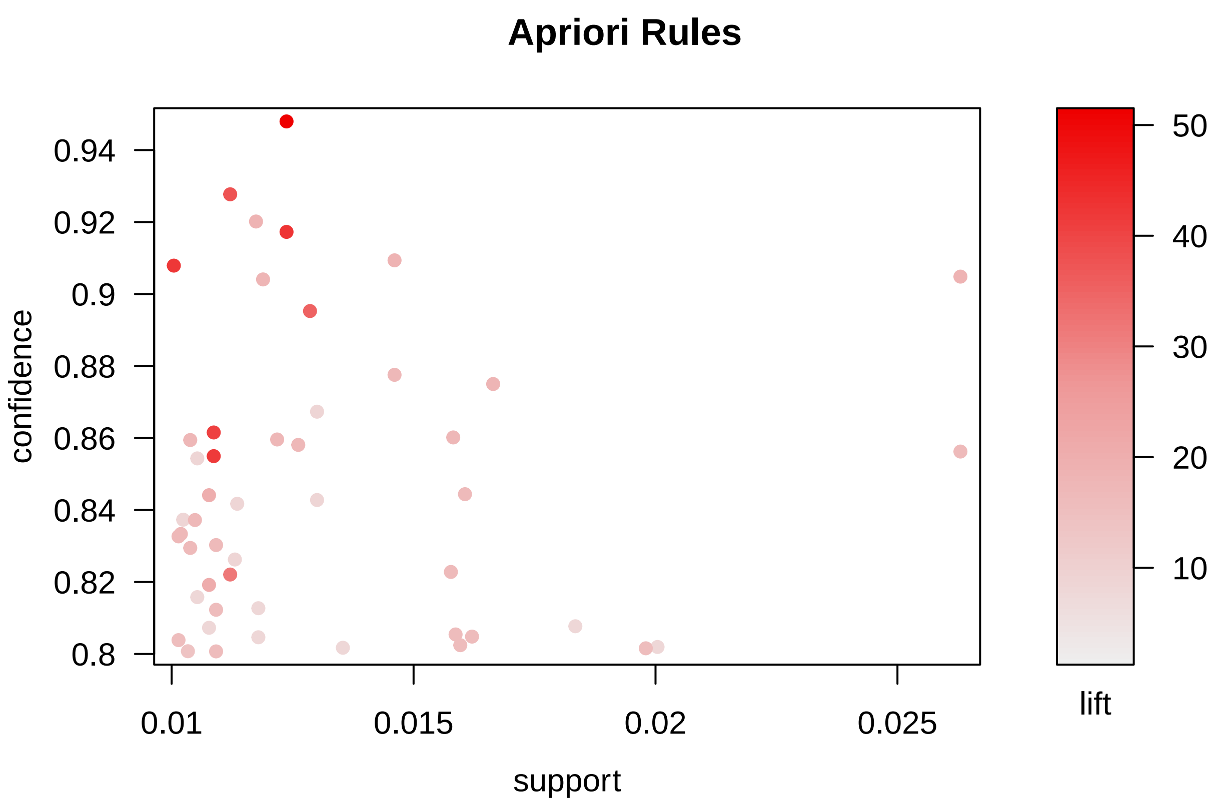
The results of the apriori algorithm are association rules which are directly seen in R Console. For ease of understanding, the rules have been converted into a data frame and the head of the data frame is shown below.

```
head(apriori.rules_df)
```

```
##                                                                lhs
## 1                  {REGENCY TEA PLATE GREEN ,REGENCY TEA PLATE PINK}
## 2                  {REGENCY TEA PLATE PINK,REGENCY TEA PLATE ROSES }
## 3          {POPPY'S PLAYHOUSE KITCHEN,POPPY'S PLAYHOUSE LIVINGROOM }
## 4         {POPPY'S PLAYHOUSE BEDROOM ,POPPY'S PLAYHOUSE LIVINGROOM }
## 5 {SET/20 RED RETROSPOT PAPER NAPKINS ,SET/6 RED SPOTTY PAPER CUPS}
## 6        {ALARM CLOCK BAKELIKE CHOCOLATE,ALARM CLOCK BAKELIKE GREEN}
##                             rhs    support confidence     lift count
## 1      {REGENCY TEA PLATE ROSES } 0.01237444  0.9172662 42.47664   255
## 2      {REGENCY TEA PLATE GREEN } 0.01237444  0.9479554 51.27170   255
## 3    {POPPY'S PLAYHOUSE BEDROOM } 0.01087009  0.8549618 41.65059   224
## 4     {POPPY'S PLAYHOUSE KITCHEN} 0.01087009  0.8615385 40.71955   224
## 5 {SET/6 RED SPOTTY PAPER PLATES} 0.01285971  0.8952703 35.00728   265
## 6     {ALARM CLOCK BAKELIKE RED } 0.01091862  0.8122744 15.92630   225
```

A scatter plot is an easy way to interpret the association rules graphically. Along the x-axis is "support", plotted against "confidence" in the y-axis, with "lift" being the 3rd dimension depicted through the color spectrum on the right of the graph

```
plot(apriori.rules, method = "scatterplot", main = "Apriori Rules")
```



FP Growth

SPACE FOR ALEX'S BLURB

ECLAT algorithm

ECLAT stands for Equivalence Class Clustering and bottom-up Lattice Traversal. The ECLAT algorithm also works on datasets that are transactions. As already seen, the dataset has been converted into "transactions" and hence the model can be readily run. The ECLAT algorithm differs from the apriori algorithm by the fact that it is a depth first algorithm, where it mines for frequent subsets by traversing the length of each branch of transaction. It starts by creating a "frequent transaction set" for each item, with a minimum support threshold. Then the function is recursively called and in each recursion, more transactions are paired and continued till there are no more transactions to pair. The main difference between Apriori & ECLAT is, in Apriori, each item is listed and paired with every other item, with Support being the stop criterion, while in ECLAT, items are listed and paired only if a corresponding transaction of that pair exists.

```
#ECLAT algorithm
eclat.itemset <- eclat(trans, parameter = list(supp=0.01,maxlen=5))
eclat.rules <- ruleInduction(eclat.itemset, trans, confidence = 0.75)
eclat.rules_df<- data.frame(lhs = labels(lhs(eclat.rules)), rhs = labels(rhs(eclat.rules)),
                             eclat.rules@quality)
```

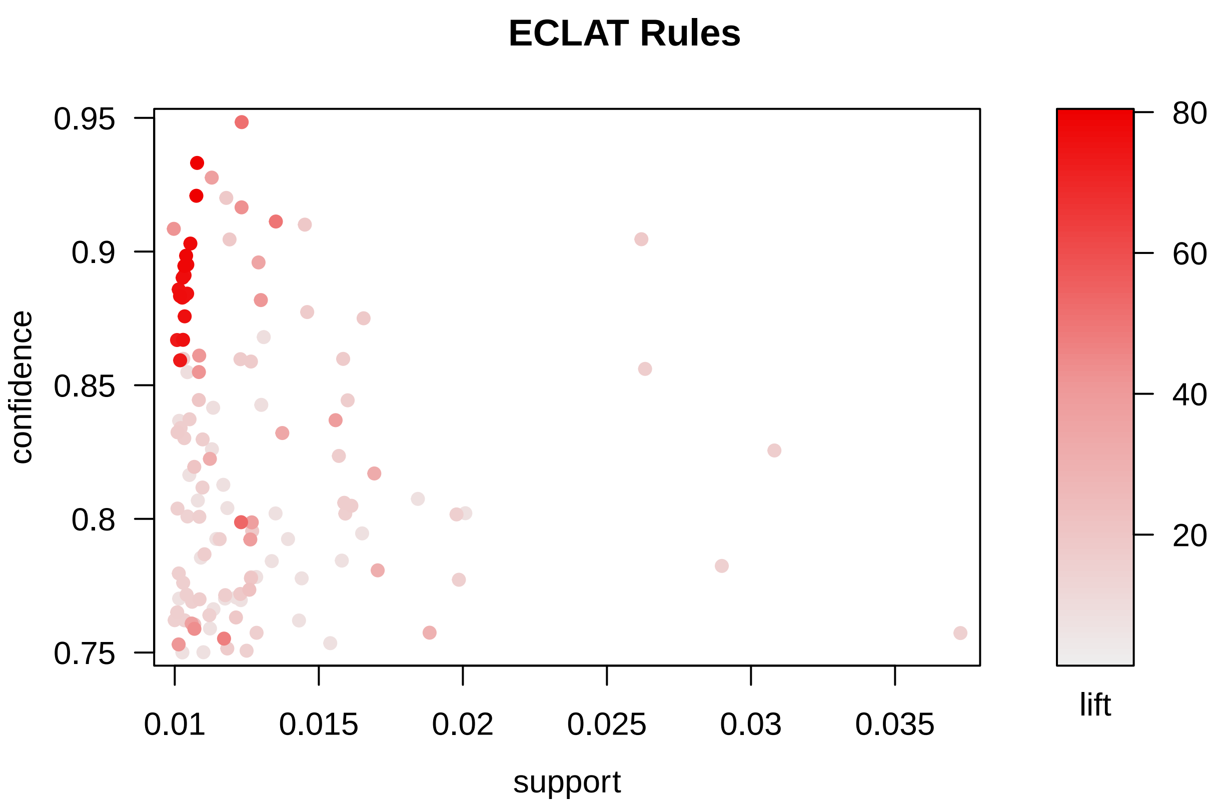
The results of the ECLAT algorithm are association rules which are directly seen in R Console. For ease of understanding, the rules have been converted into a data frame and the head of the data frame is shown below.

```r
head(eclat.rules_df)
```

```
##                                      lhs                                rhs
## 2    {CHILDRENS CUTLERY DOLLY GIRL }     {CHILDRENS CUTLERY SPACEBOY }
## 4  {CHILDRENS CUTLERY POLKADOT BLUE} {CHILDRENS CUTLERY POLKADOT PINK}
## 17              {HERB MARKER ROSEMARY}                 {HERB MARKER MINT}
## 18                  {HERB MARKER MINT}             {HERB MARKER ROSEMARY}
## 19                  {HERB MARKER MINT}                {HERB MARKER BASIL}
## 20                 {HERB MARKER BASIL}                 {HERB MARKER MINT}
##       support confidence     lift itemset
## 2  0.01072451  0.7594502 43.47219       1
## 4  0.01067598  0.7612457 37.08508       2
## 17 0.01028777  0.8833333 75.84521       9
## 18 0.01028777  0.8833333 75.84521       9
## 19 0.01009366  0.8666667 73.79917      10
## 20 0.01009366  0.8595041 73.79917      10
```

A scatter plot is an easy way to interpret the association rules graphically. Along the x-axis is "support", plotted against "confidence" in the y-axis, with "lift" being the 3rd dimension depicted through the color spectrum on the right of the graph

```r
plot(eclat.rules, method = "scatterplot", main = "ECLAT Rules")
```



Model Selection & Conclusion

The primary differentiation between Apriori & ECLAT is that the former is a "breadth-first" alogorithm and the latter is a "depth-first" algorithm. Since ECLAT is depth first, it requires less memory than Apriori algorithm. That also implies that ECLAT is a faster algorithm. To test this, the two algorithms' execution time was measured for varying support levels. It was observed that with the given dataset, there isn't a discernible difference between the two algorithms. In this case, therefore, both the algorithms are deployed in the final solution and the discretion to choose the algorithms is left to the user.

Apriori Runtime

```
ap_start <- Sys.time()
apriori.rules <- apriori(trans, parameter = list(supp=0.04, conf=0.8, maxlen=5))
apriori.rules_df<- data.frame(lhs = labels(lhs(apriori.rules)), rhs = labels(rhs(apriori.rules))
                              , apriori.rules@quality)
ap_end <- Sys.time()
ap_time <- as.numeric(ap_end - ap_start)
```

ECLAT Runtime

```
  eclat_start <- Sys.time()
  eclat.itemset <- eclat(trans, parameter = list(supp=0.04,maxlen=5))
  eclat.rules <- ruleInduction(eclat.itemset, trans, confidence = 0.8)
  eclat_end <- Sys.time()
  eclat_time <- as.numeric(eclat_end - eclat_start)
```

Table 2: Runtime comparison

Support	ap_time	eclat_time
0.01	0.403	1.826
0.02	0.623	0.697
0.03	0.271	0.562
0.04	0.320	0.504