Association Rules Mining using the Retail Market Basket Data Set

Edward Nataniel C. Apostol
2010-27627
CS 176 THX
edward.nataniel@gmail.com

1. Introduction

Association Rule Mining is a method that can be used to detect frequently occurring patterns among the items in a transaction database. It has a lot of applications in different fields. It can be used in document content analysis to detect what words frequently occur together in a text document. In marketing and sales, association rule mining is useful in supermarket shelf management. It can be used to determine what products are frequently purchased together by customers.

In this programming assignment, two of the most popular association rule mining algorithms, namely, the Apriori and FP-growth algorithms, will be used to analyze a sample retail market basket dataset. The generated results will be evaluated according to correlation measures such as Lift, Kulczynski, and Imbalance Ratio (IR).

2. Objectives

The objective of this programming assignment is to be able to demonstrate how association rule mining can be applied to a sample retail market basket dataset using two algorithms – Apriori and FP-growth. This programming assignment also aims to demonstrate the evaluation of generated association rules using Lift, Kulczynski, and Imbalance Ratio (IR).

3. Methodology

3.1 The Dataset

The dataset named *basket.dat* contains 1001 records of sample retail transactions from a grocery store. Each line represent a single transaction. Each item purchased is separated by a space character.

The R packages named *arules* and *aruleViz* were installed do the mining and visualization of the association rules.

```
library(arules)
library(arulesViz)
```

The records were read using the *read.transaction* function in the *arules* package.

```
basket <- read.transactions("basket.dat",
   format = "basket",
   sep = " ",
   rm.duplicates = TRUE)
```

3.2 Apriori Algorithm

The **Apriori** is an algorithm used to obtain frequent item sets and association rules over relational databases. The algorithm proceeds with an iterative level-wise search where k-itemsets are used to obtain (k+1)-itemsets. Apriori considers all subsets of a frequent item set to be frequent as well.

Association rules were generated using the apriori algorithm by invoking the *apriori* function in the *arules* package.

```
rules <- apriori(basket,
   parameter = list(supp = 0.1,
                    conf = 0.8))
```

The *support* of the association rule A ⇒ B, or sup(A ⇒ B), is the proportion of transactions that contains A ∪ B.

The *confidence* of the association rule A ⇒ B is the proportion transactions in which B is present whenever A is present.

$$confidence(A \Rightarrow B) = P(B|A) = \frac{sup(A \cup B)}{sup(A)}$$

A rule is considered to be strong if it satisfies the minimum support threshold and the minimum confidence threshold. For this assignment, the minimum support threshold was set at 0.1 while the minimum confidence threshold was set at 0.8.

3.2 Correlation Measures

Three correlation measures of interest were computed in this programming assignment – lift, Imbalance Ratio (IR), and Kulczynski measure.

The **lift** between the occurrence of A and B is defined as

$$Lift(A,B) = \frac{P(A \cup B)}{P(A)P(B)}$$

Note that when P(A ∪ B) = P(A)P(B), then it implies that the events A and B are independent. Hence, having a lift value of 1 means that A and B are uncorrelated. Having

a lift > 1 means that A and B are positively correlated or the occurrence of one increases the chance of occurrence of the other. When lift < 1, then A and B are negatively correlated or the occurrence of one decreases the chance of occurrence of the other. The lift of the association rule A ⇒ B is equivalent to:

$$Lift(A \Rightarrow B) = \frac{P(B|A)}{P(B)} = \frac{conf(A \Rightarrow B)}{\sup(B)}$$

The **Imbalance Ratio(IR)** assesses the imbalance of 2 itemsets A and B and is given by the formula below.

$$IR = \frac{|\sup(A) - \sup(b)|}{\sup(A) + \sup(B) - \sup(A \cup B)}$$

The range of IR is from 0 to 1. IR = 0 means that the directional implications between A and B (A ⇒ B and B ⇒ A) are the same. What we are interested in are the rules that are highly skewed or has an IR close to 1.

The **Kulczynski measure** is the arithmetic mean of the confidence measures of A and B.

$$Kulc(A,B) = \frac{1}{2}[P(A|B) + P(B|A)]$$

Since $P(B|A) = \frac{sup(A \cup B)}{sup(A)}$ and $P(A|B) = \frac{sup(A \cup B)}{sup(B)}$, the Kulczynski measure can be computed as:

$$Kulc(A,B) = \frac{1}{2}\left[\frac{sup(A \cup B)}{\sup(A)} + \frac{sup(A \cup B)}{\sup(B)}\right]$$

$$= \frac{1}{2}[conf(\mathrm{A} \Rightarrow \mathrm{B}) + \mathrm{conf}(\mathrm{B} \Rightarrow \mathrm{A})]$$

The Kulczynski measure ranges from 0 to 1. The farther the value of Kulczynski measure from 0.5, the closer the relationship between two itemsets.

In practice, IR in conjunction with Kulczynski are usually chosen as the correlation or interestingness measure.

The code in R directly calculates the three correlation measures based on the formulas given above.

```
supA = support(lhs(rules), basket,
   type = c("absolute"), control = NULL)
supB = support(rhs(rules), basket,
   type = c("absolute"), control = NULL)
supAUB = support(items(rules), basket,
   type = c("absolute"), control = NULL)

IR = abs(supA-supB)/(supA+supB-supAUB)

kulc = 0.5*((supAUB/supA)+(supAUB/supB))
```

3.2 FP-Growth Algorithm

The FP-Growth algorithm uses a divide-and-conquer strategy to compress databases into a frequent pattern tree (FP-tree). FP-Growth is faster than Aprioiri. It is efficient and scalable for mining both long and short frequent patterns.

FP-growth algorithm was executed using a C program developed by Borgelt. The program can easily be run from the command line. The general syntax of the program invocation is:

```
fpgrowth [options] infile [outfile]
```

The program was run with the ff options:

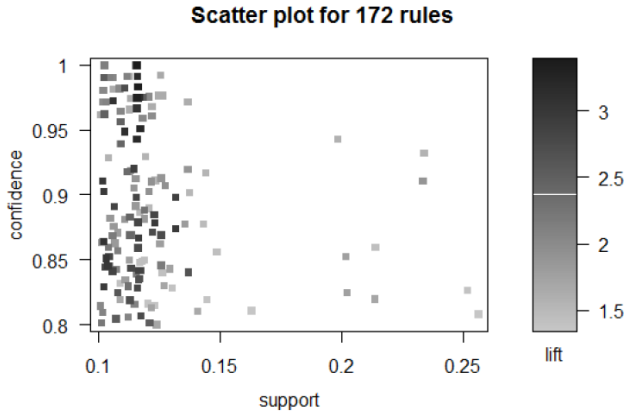
-m1: The minimum number of items per itemset is 1
-q -1: Items will be sorted according to item frequency in descending order
-s10: Minimum support is 10%
-c80: Minimum confidence is 80%
-tr : The program will generate the association rules.

The program was run twice. The first run was done to generate the frequent itemsets (without the –tr option). The second run was done to output the association rules.

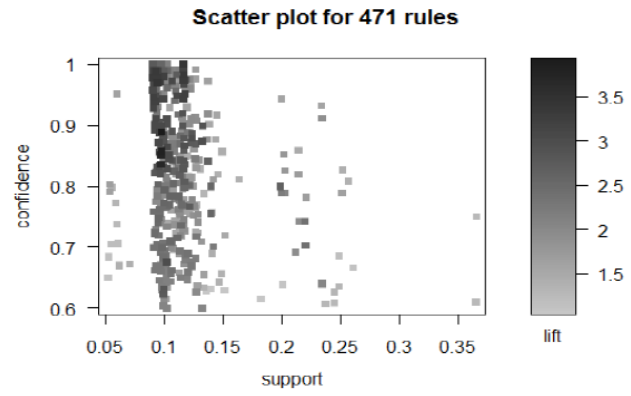
4. Results

4.1 A priori

A total of 172 rules that satisfy the minimum support threshold of 0.1 and the minimum confidence threshold of 0.8 were generated. The scatterplot of the rules was generated using the *arulesViz* package in R.



By decreasing the minimum support threshold of 0.05 and the minimum confidence threshold of 0.6, more rules were generated. The scatterplot of the rules when the thresholds are relaxed is shown below.



By inspecting the first scatterplot, it can be seen that there are very few observations that has a support value > 0.25. The rule on the rightmost side is the rule {soda}⇒{Heineken} [support=0.2567, conf=0.8082]. This means that 25.67% of all the transactions contain both soda and Heineken. For those transactions with soda, 80.82% of them also bought Heineken.

Note that the rule {Heineken}⇒{soda} did not make it to the cutoff. Heineken is present in 602 transactions but both Heineken and soda are only present in 258 transactions. Thus, the rule {Heineken}⇒{soda} didn't make the cutoff because of its low confidence of 258/602 = 42.86%.

Listed in the tables below are the top 5 rules with the highest support and top 5 rules with the highest confidence using the original threshold.

lhs	rhs	support	conf
{soda}	{heineken}	0.2567	0.8082
{artichok}	{heineken}	0.2517	0.8262
{cracker,soda}	{heineken}	0.2338	0.9323
{heineken,soda}	{cracker}	0.2338	0.9105
{baguette,hering}	{heineken}	0.2138	0.8594

Top 5 rules with the highest support

lhs	rhs	support	conf
{chicken,ice_crea,sardines}	{coke}	0.116	1
{chicken,ice_crea,sardines}	{heineken}	0.116	1
{corned_b,ham,hering,turkey}	{olives}	0.102	1
{chicken,coke,ice_crea,sardines}	{heineken}	0.116	1
{chicken,coke,heineken,sardines}	{ice_crea}	0.116	1

Top 5 rules with the highest confidence

The rule {chicken,ice_crea,sardines} => {coke} has a confidence of 1. This means that all transactions with chicken, ice cream, and sardines also have coke in them.

Generally, we want rules that have high confidence and high correlation. The correlation measures discussed earlier was computed for each rule to uncover interesting pattern relationships.

Listed in the table below are the top 5 rules with the highest lift. All five of them have a support value of 0.1159. Since the lift is >1, we can say that there is a positive correlation between the items on the LHS and RHS.

lhs	rhs	conf	lift	count
{chicken,ice_crea,sardines}	{coke}	1.000	3.382	116
{chicken,heineken,ice_crea,sardines}	{coke}	1.000	3.382	116
{chicken,coke,heineken,ice_crea}	{sardines}	1.000	3.382	116
{chicken,coke,heineken}	{sardines}	0.983	3.324	116
{chicken,heineken,sardines}	{coke}	0.975	3.297	116

Top 5 rules with the highest lift

By decreasing the minimum support threshold of 0.05 and the minimum confidence threshold of 0.6, new rules with higher lift are found.

lhs	rhs	sup	conf	lift
{apples,corned_b,hering,olives}	{steak}	1.00	1.00	3.382
{apples,corned_b,hering}	{steak}	1.000	1.000	3.382
{chicken,coke,heineken,ice_crea}	{steak}	1.000	1.000	3.382

Top 5 rules with the highest lift using a relaxed threshold

The table below by Mehmood et al.can be used to assess the Kulczynski and Imbalance Ratio (IR) values of the generated rules.

Table 13.2 Kulczynski (*Kulc*) and imbalance ratio (*IR*) values and their meanings

Relationship between LHS and RHS	Kulc/IR	Value
Positive correlation	Kulc close to 1	Kulc $\geq$ 0.7
Negative correlation	Kulc close to 0	Kulc $\leq$ 0.3
Neutral	Kulc close to 0.5	0.7 > Kulc > 0.3
Very imbalanced	IR close to 1	IR $\geq$ 0.8
Imbalanced	IR relatively close to 1	IR $\geq$ 0.6
Balanced	IR close to 0	IR $\leq$ 0.3
Neutral	Kulc and IR close to 0.5	0.6 < IR < 0.3 and 0.7 > kulc > 0.3

From the dataset, the Kulczynski values that were obtained ranges only from 0.5033 to 0.6959. So the rules were sorted by decreasing Kulczynski values and the IR of each rule were checked.

lhs	rhs	sup	conf	IR	kulc
{chicken,ice_crea, sardines}	{coke}	0.1159	1	0.6081	0.6959
{chicken,heineken, ice_crea,sardines}	{coke}	0.1159	1	0.6081	0.6959
{chicken,coke,heineken, ice_crea}	{sardines}	0.1159	1	0.6081	0.6959
{heineken,soda}	{cracker}	0.2338	0.9105	0.4521	0.6950
{chicken,coke,heineken}	{sardines}	0.1159	0.9831	0.5573	0.6875

Top 5 rules rules with the highest Kulczynski measure values

In the table above, the first three rules can be considered as interesting rules since they all have IR values that are relatively closer to 1. These rules are desired because they have a positive correlation and are imbalanced. Meanwhile, the fourth rule and fifth rules can be considered as neutral.

4.1 FP-growth

A minimum support threshold of 10 is used to generate the frequent itemset. The FP-growth program runs noticeably faster than the one using apriori. A total of 1000 items were generated. Here are the top 10 itemsets with largest support.

Item	support
Heineken	602
cracker	488
hering	486
olives	473
bourbon	403

Top 10 itemsets by largest support

The algorithm was able to find some rules that have both high support count and high confidence. Here are the 6 rules with highest support count of 116 and confidence of 100%.

lhs	rhs
{chicken,ice_crea,sardines}	{coke}
{chicken,heineken,ice_crea,sardines}	{coke}
{chicken,coke,heineken,ice_crea}	{sardines}
{chicken,coke,Heineken,sardines}	{ice_crea}
{chicken,coke,ice_cre,sardines}	{Heineken}
{chicken,ice_cre,sardines}	{Heineken}

Top 6 rules with the highest support count and confidence

The business owners may use these rules to their advantage. Supermarkets may consider these rules in their shelf management such that items are commonly bought together are close to each other. They may also offer discounts to entice the customers to buy more items.

5. Conclusion

One popular area in which Association Rule Mining can be applied is market basket analysis where consumer behavior are studied by looking at the itemsets that are frequently bought together.

There are several algorithms that can be used to mine frequent itemsets and association rules. Two of the most popular are the Apriori and the FP-growth algorithms. FP Growth is usually more efficient than Apriori.

Not all rules that will be generated are useful or interesting. That is why there are several interestingness measures such as Lift, Imbalanced Ratio (IR), and Kulczynski can be used to evaluate association rules. Lift is a good measure but it can be unstable. Using the Kulczynski measure in junction with IR measure is recommended.

The thresholds that are set such as the minimum support threshold and the minimum confidence threshold can also play a part in the rules generation. Thresholds that are too restrictive may cause the loss of some interesting rules. Thresholds that are too relaxed can generate too much rules that are non-interesting and difficult to interpret. That is why, experimenting on several combinations of threshold values are recommended.

Finally, the generated rules only serves as a guide to the decision makers of the businesses. A lot of other factors such as financial constraints may also need to be considered before implementing a specific program or recommendation. Nonetheless, association rule mining, when done right, can do wonders for businesses and consumers.

6. References

Agrawal, R., Imieliński, T., & Swami, A. (1993, June). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data* (pp. 207-216).

Borgelt, C. (2005, August). An Implementation of the FP-growth Algorithm. In *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations* (pp. 1-5).

Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.

Mehmood, R., Katib, S. S. I., & Chlamtac, I. (2020). Smart Infrastructure and Applications. Springer International Publishing.

Naval, P. (2015). Association Rules (Lecture 4). CS 176 Course Materials. UP Department of Computer Science.