STA 380 Homework 2

Wenduo Wang

August 12, 2016

Import `arules` library for association rule mining.

```
library(arules)
```

Import data with `read.transactions()` function from `arules`, which will automatically convert each row into a list of items separated by commas, and the returned object is a `transactions` object. This function will also drop duplicate items from each basket if `rm.duplicates = TRUE`.

```
groceries <- read.transactions("data/groceries.txt", sep=",", rm.duplicates = TRUE)
```

We can assign each user an id by converting the `transactions` to a list with `as(from="transactions", to="list")`, and define `names()` of the list, and then convert the list back to `transactions`.

```
groceries_list <- as(groceries, "list")
names(groceries_list) <- as.character(1:length(groceries_list))
groceries <- as(groceries_list, "transactions")
```

At this point, the *groceries* object is suitable for a priori analysis. Before applying the `apriori` function, we need to determine what the *support* and *confidence* thresholds, and *maxlen* value. This is essentially a heuristic process, so here let's first try a higher *support* level 0.01 and *confidence* threshold 0.55 (just a little bit more than 0.5), and see what we get.

```
params <- list(support=.01, confidence=.55, maxlen=4)
grocery_rules <- apriori(groceries, parameter = params)
```

```
> Apriori
>
> Parameter specification:
>  confidence minval smax arem  aval originalSupport support minlen maxlen
>        0.55    0.1    1 none FALSE            TRUE    0.01      1      4
>  target   ext
>   rules FALSE
>
> Algorithmic control:
>  filter tree heap memopt load sort verbose
>     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
>
> Absolute minimum support count: 98
>
> set item appearances ...[0 item(s)] done [0.00s].
> set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
> sorting and recoding items ... [88 item(s)] done [0.00s].
> creating transaction tree ... done [0.00s].
> checking subsets of size 1 2 3 4 done [0.00s].
> writing ... [7 rule(s)] done [0.00s].
> creating S4 object  ... done [0.00s].
```

```
inspect(subset(grocery_rules, subset=lift>=2))
```

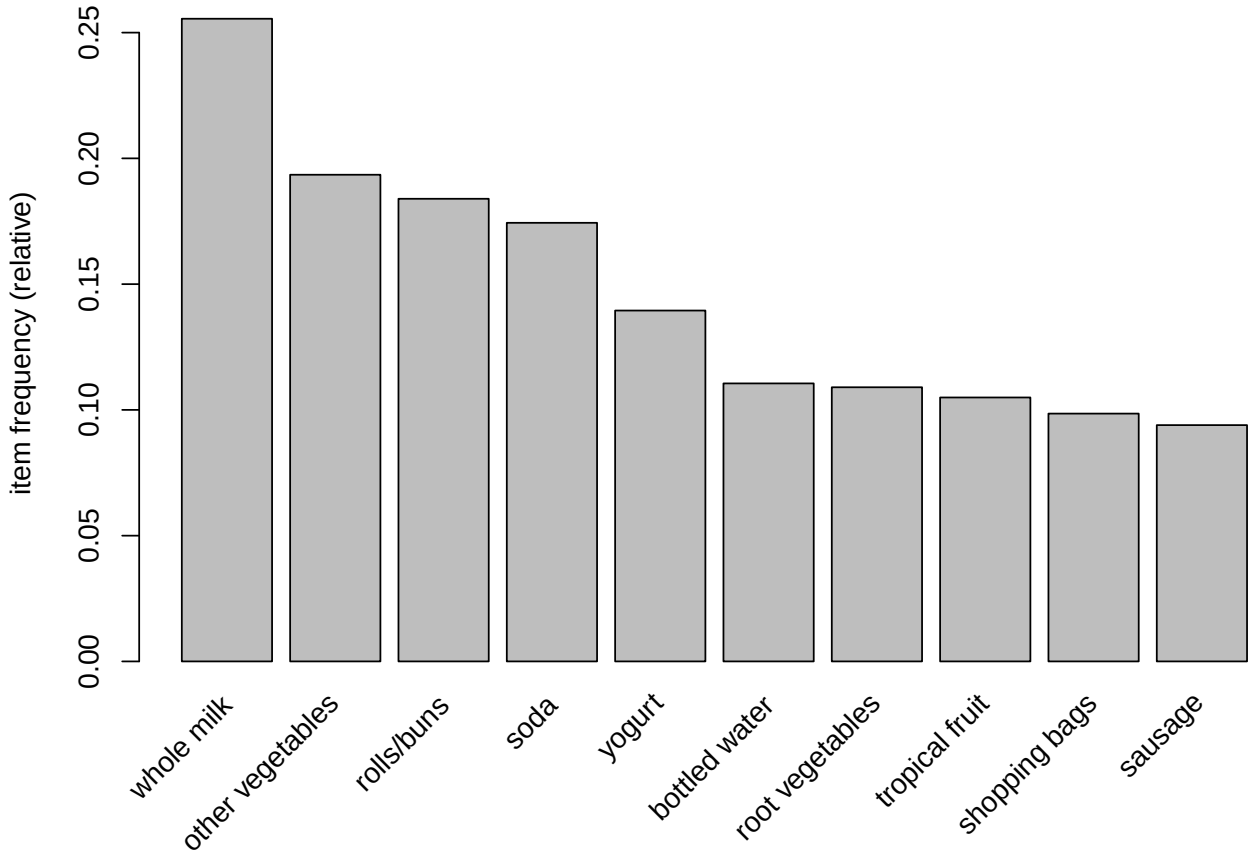
```
>   lhs                  rhs                  support confidence     lift
> 1 {curd,
```

```
>    yogurt}           => {whole milk}       0.01006609  0.5823529 2.279125
> 2 {butter,
>    other vegetables} => {whole milk}       0.01148958  0.5736041 2.244885
> 3 {domestic eggs,
>    other vegetables} => {whole milk}       0.01230300  0.5525114 2.162336
> 4 {citrus fruit,
>    root vegetables}  => {other vegetables} 0.01037112  0.5862069 3.029608
> 5 {root vegetables,
>    tropical fruit}   => {other vegetables} 0.01230300  0.5845411 3.020999
> 6 {root vegetables,
>    tropical fruit}   => {whole milk}       0.01199797  0.5700483 2.230969
> 7 {root vegetables,
>    yogurt}           => {whole milk}       0.01453991  0.5629921 2.203354
```

Here we only selected the associations with *lift* greater than 2, which gives 7 in total. And among them are items such as *other vegetables* and *whole milk*, which are themselves frequent terms across all baskets. Such results provide limited information, so we need to look closer into more interesting and less ubiquitous items.

So a natural question to be asked here is, which are the most frequent items? Let's make a plot to show the top 10 frequent terms.

```
itemFrequencyPlot(groceries, topN=10)
```



So here we can see clearly that *whole milk* and *other vegetables* are indeed frequent terms containing relatively less information.

Therefore, let's lower *support* level to 0.001 to include less often items. Also, when printing out the associations, we raise the *lift* threshold to 10, which indicates highly correlated and dependent items.

```
params <- list(support=.001, confidence=.55, maxlen=4)
grocery_rules <- apriori(groceries, parameter = params)
```

```
> Apriori
>
> Parameter specification:
>  confidence minval smax arem  aval originalSupport support minlen maxlen
>        0.55    0.1    1 none FALSE            TRUE   0.001      1      4
>  target   ext
>   rules FALSE
>
> Algorithmic control:
>  filter tree heap memopt load sort verbose
>     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
>
> Absolute minimum support count: 9
>
> set item appearances ...[0 item(s)] done [0.00s].
> set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
> sorting and recoding items ... [157 item(s)] done [0.00s].
> creating transaction tree ... done [0.00s].
> checking subsets of size 1 2 3 4 done [0.01s].
> writing ... [3314 rule(s)] done [0.00s].
> creating S4 object  ... done [0.00s].
```

```
inspect(subset(grocery_rules, subset=lift>=10))
```

```
>   lhs                         rhs                  support confidence     lift
> 1 {liquor,
>    red/blush wine}        => {bottled beer}   0.001931876  0.9047619 11.23527
> 2 {popcorn,
>    soda}                  => {salty snack}    0.001220132  0.6315789 16.69779
> 3 {Instant food products,
>    soda}                  => {hamburger meat} 0.001220132  0.6315789 18.99565
> 4 {ham,
>    processed cheese}      => {white bread}    0.001931876  0.6333333 15.04549
> 5 {baking powder,
>    flour}                 => {sugar}          0.001016777  0.5555556 16.40807
> 6 {hard cheese,
>    whipped/sour cream,
>    yogurt}                => {butter}         0.001016777  0.5882353 10.61522
> 7 {hamburger meat,
>    whipped/sour cream,
>    yogurt}                => {butter}         0.001016777  0.6250000 11.27867
```

Here we see some intriguing associations which are less frequent among all baskets but exhibits huge correlation in terms of *lift*, which is a measure of dependence. While in the previous case there were only 7 associations even with *lift* level higher than 2, here there are 7 associations with *lift* higher than 10.

Let's look at the association {liquor, red/blush wine} => {bottled beer}, it is intuitively this is some combination appealing to an alcohol lover. This intuition also holds for other association groups, such as {baking powder, flour} => {sugar} which is probably a part of common baking recipe.

And what about other associations?

```
params <- list(support=.001, confidence=.55, maxlen=4)
grocery_rules <- apriori(groceries, parameter = params)
```

```
> Apriori
>
> Parameter specification:
>  confidence minval smax arem  aval originalSupport support minlen maxlen
>        0.55    0.1    1 none FALSE            TRUE   0.001      1      4
>  target   ext
>   rules FALSE
>
> Algorithmic control:
>  filter tree heap memopt load sort verbose
>     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
>
> Absolute minimum support count: 9
>
> set item appearances ...[0 item(s)] done [0.00s].
> set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
> sorting and recoding items ... [157 item(s)] done [0.00s].
> creating transaction tree ... done [0.00s].
> checking subsets of size 1 2 3 4 done [0.01s].
> writing ... [3314 rule(s)] done [0.00s].
> creating S4 object  ... done [0.00s].
```

```
inspect(subset(grocery_rules, subset=(lift<=10 & lift>8)))
```

```
>    lhs                        rhs                        support confidence     lift
> 1  {frozen vegetables,
>     specialty chocolate}   => {fruit/vegetable juice} 0.001016777  0.6250000 8.645394
> 2  {frozen fish,
>     other vegetables,
>     tropical fruit}        => {pip fruit}             0.001016777  0.6666667 8.812724
> 3  {flour,
>     root vegetables,
>     whole milk}            => {whipped/sour cream}    0.001728521  0.5862069 8.177794
> 4  {misc. beverages,
>     other vegetables,
>     tropical fruit}        => {fruit/vegetable juice} 0.001016777  0.5882353 8.136841
> 5  {citrus fruit,
>     fruit/vegetable juice,
>     grapes}                => {tropical fruit}        0.001118454  0.8461538 8.063879
> 6  {fruit/vegetable juice,
>     grapes,
>     tropical fruit}        => {citrus fruit}          0.001118454  0.6875000 8.306588
> 7  {citrus fruit,
>     grapes,
>     tropical fruit}        => {fruit/vegetable juice} 0.001118454  0.6111111 8.453274
> 8  {butter,
>     hard cheese,
>     yogurt}                => {whipped/sour cream}    0.001016777  0.6250000 8.718972
> 9  {butter,
>     hard cheese,
>     other vegetables}      => {whipped/sour cream}    0.001220132  0.6000000 8.370213
> 10 {butter,
>     hard cheese,
```

```
>     whole milk}           => {whipped/sour cream}    0.001423488  0.6666667 9.300236
> 11 {ham,
>     other vegetables,
>     tropical fruit}       => {pip fruit}             0.001626843  0.6153846 8.134822
> 12 {butter,
>     sliced cheese,
>     whole milk}           => {whipped/sour cream}    0.001220132  0.6000000 8.370213
> 13 {cream cheese,
>     sugar,
>     whole milk}           => {domestic eggs}         0.001118454  0.5500000 8.668670
> 14 {curd,
>     sugar,
>     yogurt}               => {whipped/sour cream}    0.001016777  0.6250000 8.718972
> 15 {butter,
>     other vegetables,
>     sugar}                => {whipped/sour cream}    0.001016777  0.7142857 9.964539
> 16 {citrus fruit,
>     cream cheese,
>     whole milk}           => {domestic eggs}         0.001626843  0.5714286 9.006410
> 17 {domestic eggs,
>     frankfurter,
>     tropical fruit}       => {pip fruit}             0.001016777  0.6250000 8.261929
> 18 {shopping bags,
>     tropical fruit,
>     whipped/sour cream}   => {pip fruit}             0.001118454  0.6470588 8.553526
```

Above are associations with *lift* between 8 and 10. And here we can see some interesting combinations such as {butter, hard cheese, milk} => {whipped/sour cream}. Why is the customer buying such protein and fat heavy foots altogether? Probably it is simply because of the way these products are placed in the store. If some products are placed together, then they are more likely to be sold in a bundle. This can also be seen in {citrus fruit, grapes, tropical fruit} => {fruit/vegetable juice}.