Lab

Ref: http://cran.csiro.au/web/packages/arules/vignettes/arules.pdf [accessed 04/11/2022]

Aim

To learn about association rules, including the *apriori* algorithm and association rule visualisation.

Before you start

Load the following packages (download them if needed):

- arules - it contains the *apriori* algorithm and the *Groceries* dataset
- arulesViz – it contains visualisation facilities for association rules. You may need to download and install associated packages too!

In this lab you will be using the *Groceries* dataset which is in package *arules*.

```
data("Groceries")
```

```
summary(Groceries)
```

It contains 9835 instances of market basket transactions described by 169 attributes.

Algorithm *apriori*

The *apriori* algorithm is in the *arules* package. It offers the option of specifying the minimum support and confidence. Apply this algorithm to the *Groceries* dataset with a minimum support of 0.1% and a minimum confidence of 50%

```
rules <- apriori(Groceries, parameter=list(support=0.001, confidence=0.5))
```

This produces 5668 rules. There are too many rules to check them all, but you can check the first few, after ordering them (e.g. by lift or by confidence). Try the following code to select the first 10 rules, sorted by lift.

```
inspect(head(sort(rules, by ="lift"),10))
```

The following code to select the first 15 rules, sorted by confidence.

```
inspect(head(sort(rules, by ="confidence"),15))
```

Exercise 1

Compare the first 4 rules returned if sorting the rules by lift with those returned if sorting the rules by confidence. Explain any discrepancies in the results.

Rule quality

You can inspect the quality of rules by

```
quality(rules)
```

This displays the support, confidence and lift of rules. There are too many, so to find out the quality of the first few

```
head(quality(rules))
```

Removing redundant rules

Some rules may be redundant (covered by other rules). You may remove them, using a quality measure (e.g. lift) as follows:

```
redundant <- is.redundant(rules, measure = "lift")

# remove redundant rules - i.e. exclude rules in "redundant" from set

rules <- rules[!redundant]
```

Plotting rules

It is often useful to visualise the resulting rules using a plot.

```
plot(rules)
```

This is equivalent to

```
plot(rules, measure = "support", shading = "lift")
```

This plots the rules' support against its confidence. The lift of a rule is represented by the intensity of the colour. The more intense the colour, the higher the lift.

You could change the plot to showing support against lift and have confidence as the intensity in the colour. Generally rules with high lift have low support.

```
plot(rules, measure=c("support", "lift"), shading="confidence")
```

You could also use the shading to show the number of items in the itemsets, i.e. the "order" of the rule.

```
plot(rules, shading="order")
```

You can see that the higher the order, the lower the support, i.e. large itemsets occur less often that smaller ones.

To include a title for you plot you can use `main` in the control list as follows

```
plot(rules, shading="order",
      control=list(main = "Rule size - Support vs confidence"))
```

Exercises

2. Plot the rule set with support against lift, using the itemset size for the shading.
3. Plot the rule set with confidence against lift, using the itemset size for the shading. Can you see any pattern?

Interactive plots

The plots for visualising association rules can be interactive. Interactive features include:

1. Checking single rules by selecting them and clicking the inspect button.
2. Checking groups of rules by selecting a rectangular region of the plot and clicking the inspect button.
3. Zooming into a selected region (zoom in/zoom out buttons).
4. Filtering rules. The measure used for shading is used for the filtering.
5. Selecting a cut-off point for the shading measure. All rules with a measure lower than the cut-off point will be filtered.
6. Returning the last selection for further analysis (end button).

In order to obtain an interactive plot, try the following **in the console window. It will not work if run within an Rmd file.** Note how the result of plotting is assigned to a variable so that the returned selection can be analysed.

```
plot(rules, measure=c("support", "lift"),
      shading="confidence", engine ="interactive")
```

If you have trouble with this plot (pointer offset issues), skip to the "Alternative interactive plot" section below.

You will first need to select a region by clicking on 2 different points. The tool will form a rectangle using the line between the 2 points as the diagonal. You can then apply functions to the rules in the selected region (rectangle). The functions are *inspect, filter, zoom* in and *zoom out*. There is also the option *end*, to stop the interactive plot.

Alternative interactive plots

Interactive plots can also be plotted using an *htmlwidget***.** The command for this is:

```
plot(rules, measure=c("support", "lift"),
      shading="confidence", engine ="htmlwidget")
```

The widget lets you explore the plot and mouseover any of the points to see the rules. You can also zoom in on the plot by selecting a rectangle (click and drag or use the UI) and zoom out (double click on the plot). But you cannot filter the points or change the shading from within the plot.

Rule selection

You may wish to only plot selected rules, for example, with a minimum confidence of 75%

```
confrules <- rules[quality(rules)$confidence >= 0.75]
plot(confrules)
```

Exercises

4. Plot rules with a minimum confidence of 70% and a minimum lift of 6.
5. It is often the case that the most interesting rules are in the support/confidence border. Obtain an interactive support vs. confidence plot and inspect the border. Are there any interesting rules?

Matrix plots

Matrix plots can be used to see the rules' LHS (left hand side - conditions) against their RHS (right hand side – conclusions). Try the following for rules with at least 80% confidence.

```
confrules <- rules[quality(rules)$confidence >= 0.8]
plot(confrules, method="matrix", measure="lift")
```

Reordering rules may give better visuals: it may bring rules with similar values for the interest measure closer together. Try

```
plot(confrules, method="matrix", measure="lift",
            control=list(reorder='support/confidence'))
```

Options for reordering are *'none', 'support/confidence', 'measure'* or *'similarity'*.

Remember, you can always add `engine = "htmlwidget"` to get an interactive plot which might make interpreting the graphs substantially easier (i.e. get the mous eover functionality).

Exercise 6

Plot (and explore) a matrix of rules with a lift of at least 7. Use confidence as your measure.

3D Matrices

Instead of a standard matrix, you may use a 3D one. Try

```
plot(confrules, method="matrix3D", measure="lift")
```

Reordering - options are 'none', 'support/confidence', 'measure' or 'similarity'. Using option 'support/confidence'.

```
plot(confrules, method="matrix3D", measure="lift",
            control=list(reorder='support/confidence'))
```

Using 2 measures together

You can plot 2 measures together, one using colour and the other using luminance. Try.

```
plot(confrules, method="matrix", measure=c("lift", "confidence"))
```

and reordering for changed visuals

```
plot(confrules, method="matrix", measure=c("lift", "confidence"),
control=list(reorder='similarity'))
```

Parallel coordinates plots

These plots are designed to visualize multidimensional data where each dimension is displayed separately on the x-axis and the y-axis is shared. Each data point is represented by a line connecting the values for each dimension.

An arrow is used for representation. The arrow's head points to the consequent item. The arrow spans enough positions on the x-axis to represent all the items in the rule, so arrows are longer if they represent rules with more items.

The width of the arrows represents its support and the colour intensity its confidence.

For example, try the following for the best 10 rules for lift.

```
liftrules <- head(sort(rules, by="lift"), 10)
```

```
plot(liftrules, method="paracoord")
```

And with reordering of rules

```
plot(liftrules, method="paracoord", control=list(reorder=TRUE))
```

Exercises

7. Select the best 100 rules according to support
8. Keep only those with a confidence of 60% and a lift of 3.
9. Plot the rules. Experiment with various plotting options. Use the interactive engine (or htmlwidget) to visualise each rule.
10. Repeat the exercise above but reduce the lift of 2.
    a. Do you get more or less rules?
    b. Compare the maximum support obtained with the one in the previous exercise.
    c. Compare the maximum confidence with the one obtained in the previous exercise.

Selecting rules with specific conditions/conclusions

Assume that you are interested in rules regarding butter or white bread as conclusions (i.e. in the right hand side of the rule – rhs). You may focus apriori on the generation of rules which conclude whether sugar is bought the appearance field in the call to the apriori algorithm. For example

```
butterWhiteBread <- apriori(Groceries,
      parameter=list(support=0.001, confidence=0.5),
      appearance = list(rhs=c("butter", "white bread"), default="lhs"),
      control = list(verbose=F))

inspect(head(sort(butterWhiteBread, by ="lift"),15))
```

Note that you may want to remove redundant rules before displaying the results in some cases.

Similarly, we may wish to find rules with butter or white bread in the condition (lhs) of the rule which conclude whole milk

```
butterWhiteBreadWholeMilk <- apriori(Groceries,
                  parameter=list(support=0.001, confidence=0.5),
                  appearance = list(lhs=c("butter", "white bread"),
                        rhs=c("whole milk")),
                  control = list(verbose=F))

inspect(head(sort(butterWhiteBreadWholeMilk, by ="lift"),15))
```

Exercise

Experiment with rule generation using other conditions or conclusions (i.e. other grocery products.

Be curious, find patterns empirically and justify them with the theory you observed in the lecture.

- What happens to the number of rules when you increase the confidence threshold?
- What happens to the number of rules when you increase the minimum support?
- If you order the rules, are the "top" rules the same?
- What values are required to get a minimum of 5 rules?
- What happens if you set a support of 0?
- Can you find an empty itemset?
- What do you think this means?