Association Rules

Solon Karapanagiotis

30 Jul 2016

Contents

Introduction **1**

Methods **2**

Results **7**

Discussion **8**

References **8**

The `arules` and `arulesViz` package need to be installed and loaded.

The purpose is to describe the associations and patterns among the set of input variables (unsupervised learning problem).

Introduction

The data were collected from students enrolled in an introductory statistics course at a large university in the US over a four year period. An opening course survey was administered to the students. The anonymous survey was available on the course website and contained questions on student demographic variables, such as gender, height, eye color, whether or not a student exercises and for how many hours per week, etc. After seven semesters, the full data set contains 2,068 records on 14 different categorical and quantitative variables (Froelich and Stephenson 2013).

```
str(eyecolorgenderdata)
```

```
## 'data.frame':    2068 obs. of  14 variables:
##  $ gender       : Factor w/ 2 levels "female","male": 1 2 1 2 1 2 2 2 1 2 ...
##  $ age          : int  18 20 18 23 19 19 37 22 26 21 ...
##  $ year         : Factor w/ 6 levels "first","first\"",..: 1 6 1 3 5 5 6 6 3 6 ...
##  $ eyecolor     : Factor w/ 5 levels "blue","brown",..: 4 2 3 4 1 3 2 2 2 1 ...
##  $ height       : int  68 70 67 74 62 67 74 73 70 68 ...
##  $ miles        : num  195 120 200 140 60 0 511 210 120 90 ...
##  $ brothers     : int  0 3 0 1 0 0 1 3 2 1 ...
##  $ sisters      : int  1 0 1 1 1 1 1 2 1 1 ...
##  $ computertime : num  20 24 35 5 5 5 3 2 5 5 ...
##  $ exercise     : Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 1 2 1 2 ...
##  $ exercisehours: num  3 0 3 25 4 8 0 10 0 3 ...
##  $ musiccds     : int  75 50 53 50 30 100 50 100 130 34 ...
##  $ playgames    : num  6 0 8 0 2 0 3 6 0 0.5 ...
##  $ watchtv      : num  18 3 1 7 5 10 8 10 20 5 ...
```

Methods

I implement Association Rules based on Agrawal, Srikant, and others (1994).

Short background

An association rule is in the form of $A \Rightarrow B$, where A and B are two disjoint itemsets, referred to respectively as the lhs (left-hand side) and rhs (right-hand side) of the rule. The three most widely-used measures for selecting interesting rules are *support*, *confidence* and *lift*.

- *support*: the percentage of cases in the data that contains both A and B [$support(A \Rightarrow B) = P(A \cup B)$],
- *confidence*: the percentage of cases containing A that also contain B [$confidence(A \Rightarrow B) = [P(B|A) = \frac{P(A \cup B)}{P(A)}$],
- *lift*: the ratio of confidence to the percentage of cases containing B [$lift(A \Rightarrow B) \frac{confidence(A \Rightarrow B)}{P(B)} = \frac{P(A \cup B)}{P(A)P(B)}$].

where $P(A)$ is the percentage (or probability) of cases containing A.

I implement the APRIORI algorithm here (Agrawal, Srikant, and others 1994). Median splits are used for the quantitative variables.

```
eyecolorgenderdata[["age"]] <- factor((as.numeric(eyecolorgenderdata[["age"]]) >
                                         median(eyecolorgenderdata$age,  na.rm = T)) + 1,
                                       levels = 1 : 2 , labels = c("younger", "older"))

eyecolorgenderdata[["height"]] <- factor((as.numeric(eyecolorgenderdata[["height"]]) >
                                            median(eyecolorgenderdata$height, na.rm = T))+1,
                                          levels = 1 : 2 , labels = c("taller", "shorter"))

eyecolorgenderdata[["miles"]] <- factor((as.numeric(eyecolorgenderdata[["miles"]]) >
                                           median(eyecolorgenderdata$miles, na.rm = T))+1,
                                         levels = 1 : 2 , labels = c("closer", "away"))

eyecolorgenderdata[["computertime"]] <- factor((as.numeric(eyecolorgenderdata[["computertime"]]) >
                                                  median(eyecolorgenderdata$computertime,
                                                         na.rm = T))+1,
                                                levels = 1 : 2 , labels = c("compLow", "compHigh"))

eyecolorgenderdata[["exercisehours"]] <- factor((as.numeric(eyecolorgenderdata[["exercisehours"]]) >
                                                   median(eyecolorgenderdata$exercisehours,
                                                          na.rm = T))+1,
                                                 levels = 1 : 2 , labels = c("exercLow", "exercHigh"))

eyecolorgenderdata[["musiccds"]] <- factor((as.numeric(eyecolorgenderdata[["musiccds"]]) >
                                              median(eyecolorgenderdata$musiccds, na.rm = T))+1,
                                            levels = 1 : 2 , labels = c("musicLow", "musicHigh"))

eyecolorgenderdata[["playgames"]] <- factor((as.numeric(eyecolorgenderdata[["playgames"]]) >
                                               median(eyecolorgenderdata$playgames,  na.rm = T))+1,
                                             levels = 1 : 2 , labels = c("gamesLow", "gamesHigh"))

eyecolorgenderdata[["watchtv"]] <- factor((as.numeric(eyecolorgenderdata[["watchtv"]]) >
```

```
                                                median(eyecolorgenderdata$watchtv, na.rm = T))+1,
                                              levels = 1 : 2 , labels = c("TVLow", "TVHigh"))

eyecolorgenderdata[["brothers"]] <- as.factor(eyecolorgenderdata[["brothers"]])

eyecolorgenderdata[["sisters"]] <- as.factor(eyecolorgenderdata[["sisters"]])

eyecolorgenderdata.raw <- eyecolorgenderdata
```

I use the `apriori()` function for the association rule mining.

```
# all rules
rules.all <- apriori(eyecolorgenderdata.raw, control = list(verbose=F),
                 parameter = list(minlen=2, maxlen=10, supp=0.005, conf=0.8))
rules.all
```

```
## set of 520013 rules
```

- The *support*, *confidence* and the *min-maximum length* of rules were set to 0.005, 0.8, 2-10, respectively
  - The minimum support (`supp`) of 0.005, implies that each rule is supported at least by 11 (=ceiling(0.005*2068)) cases, which is acceptable for a sample of 2,068.
  - min (`minlen`) and max (`maxlen`) length of rules: the min length is set to 2 to avoid the left-hand side (lhs) of any rule to be empty.
- the details of progress are suppressed with `verbose=F`.

These specifications create many uninteresting rules (520013 in total!). To make things more interpretable I focus on rules with indicating the eye colour (blue, brown, green, hazel, other) as consequent (right-hand side).

```
rules <- apriori(eyecolorgenderdata.raw, control = list(verbose=F),
                 parameter = list(minlen=2, maxlen=10, supp=0.005, conf=0.8),
                 appearance = list(rhs=c("eyecolor=blue", "eyecolor=brown", "eyecolor=green",
                                         "eyecolor=hazel", "eyecolor=other"), default="lhs"))
```

Since I am interested in only rules with rhs indicating eyecolor, I set `rhs=c("eyecolor=blue", "eyecolor=brown", "eyecolor=green", "eyecolor=hazel", "eyecolor=other")`. All other items can appear in the lhs, as set with `default="lhs"`. After association rule mining, rules are sorted by lift to make high-lift rules appear first.

```
rules.sorted <- sort(rules, by="lift")
inspect(head(rules.sorted)) #prints the first 6 rules
```

```
##   lhs                           rhs                  support confidence     lift
## 1 {gender=male,
##    sisters=0,
##    computertime=compHigh,
##    exercise=No,
##    musiccds=musicLow,
##    playgames=gamesLow}      => {eyecolor=brown} 0.005319149  0.9166667 2.952752
## 2 {gender=male,
```

```
##    sisters=0,
##    computertime=compHigh,
##    exercise=No,
##    exercisehours=exercLow,
##    musiccds=musicLow,
##    playgames=gamesLow}      => {eyecolor=brown} 0.005319149  0.9166667 2.952752
## 3 {age=older,
##    height=taller,
##    brothers=1,
##    computertime=compLow,
##    exercisehours=exercLow,
##    playgames=gamesHigh}     => {eyecolor=brown} 0.005802708  0.8571429 2.761015
## 4 {height=taller,
##    miles=closer,
##    sisters=2,
##    exercisehours=exercHigh,
##    playgames=gamesHigh}     => {eyecolor=brown} 0.005319149  0.8461538 2.725617
## 5 {height=taller,
##    miles=closer,
##    sisters=2,
##    exercise=Yes,
##    exercisehours=exercHigh,
##    playgames=gamesHigh}     => {eyecolor=brown} 0.005319149  0.8461538 2.725617
## 6 {age=older,
##    brothers=1,
##    computertime=compLow,
##    exercise=No,
##    musiccds=musicHigh,
##    playgames=gamesHigh}     => {eyecolor=brown} 0.005319149  0.8461538 2.725617
```

Some rules generated provide little or no extra information when some other rules are in the result. For example, the above rule 2 provides no extra knowledge in addition to rule 1, since rules 1 tells us that whatever the exercisehours the eye color is brown. Generally speaking, when a rule (such as rule 2) is a super rule of another rule (such as rule 1) and the former has the same or a lower lift, the former rule (rule 2) is considered to be redundant. Other redundant rules in the above result are rules 5, 7, 11, 16, 19 and 21.

```
# find redundant rules
subset.matrix <- is.subset(rules.sorted, rules.sorted) # function is.subset(r1, r2) checks whether r1 i
subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA
redundant <- colSums(subset.matrix, na.rm=T) >= 1
which(redundant)
```

```
##      {gender=male,eyecolor=brown,sisters=0,computertime=compHigh,exercise=No,exercisehours=exercLow,r
##
##                              {eyecolor=brown,height=taller,miles=closer,sisters=2,exercise=Yes,exercis
##
##      {age=older,eyecolor=brown,brothers=1,computertime=compLow,exercise=No,exercisehours=exercLow,mus
##
##                                          {gender=male,year=first,eyecolor=brown,miles=away,brothers=
##
##                    {gender=female,eyecolor=brown,height=taller,miles=closer,brothers=0,sisters=1,exer
##
## {eyecolor=blue,miles=away,brothers=1,sisters=1,computertime=compLow,exercise=Yes,exercisehours=exercl
##
```

```
##                     {age=younger,year=first,eyecolor=blue,sisters=1,exercise=Yes,exercisehours=exercl
##
```

Below I prune redundant rules. Note that the rules have already been sorted descendingly by lift.

```
# remove redundant rules
rules.pruned <- rules.sorted[!redundant]
inspect(rules.pruned)
```

```
##    lhs                           rhs                  support confidence     lift
## 1  {gender=male,
##     sisters=0,
##     computertime=compHigh,
##     exercise=No,
##     musiccds=musicLow,
##     playgames=gamesLow}      => {eyecolor=brown} 0.005319149  0.9166667 2.952752
## 2  {age=older,
##     height=taller,
##     brothers=1,
##     computertime=compLow,
##     exercisehours=exercLow,
##     playgames=gamesHigh}     => {eyecolor=brown} 0.005802708  0.8571429 2.761015
## 3  {height=taller,
##     miles=closer,
##     sisters=2,
##     exercisehours=exercHigh,
##     playgames=gamesHigh}     => {eyecolor=brown} 0.005319149  0.8461538 2.725617
## 4  {age=older,
##     brothers=1,
##     computertime=compLow,
##     exercise=No,
##     musiccds=musicHigh,
##     playgames=gamesHigh}     => {eyecolor=brown} 0.005319149  0.8461538 2.725617
## 5  {miles=away,
##     brothers=3,
##     computertime=compLow,
##     watchtv=TVHigh}          => {eyecolor=brown} 0.006286267  0.8125000 2.617212
## 6  {gender=male,
##     year=first,
##     miles=away,
##     brothers=0,
##     exercise=No}             => {eyecolor=brown} 0.005802708  0.8000000 2.576947
## 7  {height=taller,
##     miles=closer,
##     sisters=2,
##     exercise=Yes,
##     playgames=gamesHigh}     => {eyecolor=brown} 0.005802708  0.8000000 2.576947
## 8  {age=older,
##     brothers=1,
##     sisters=0,
##     computertime=compLow,
##     exercisehours=exercLow,
##     playgames=gamesHigh}     => {eyecolor=brown} 0.005802708  0.8000000 2.576947
```

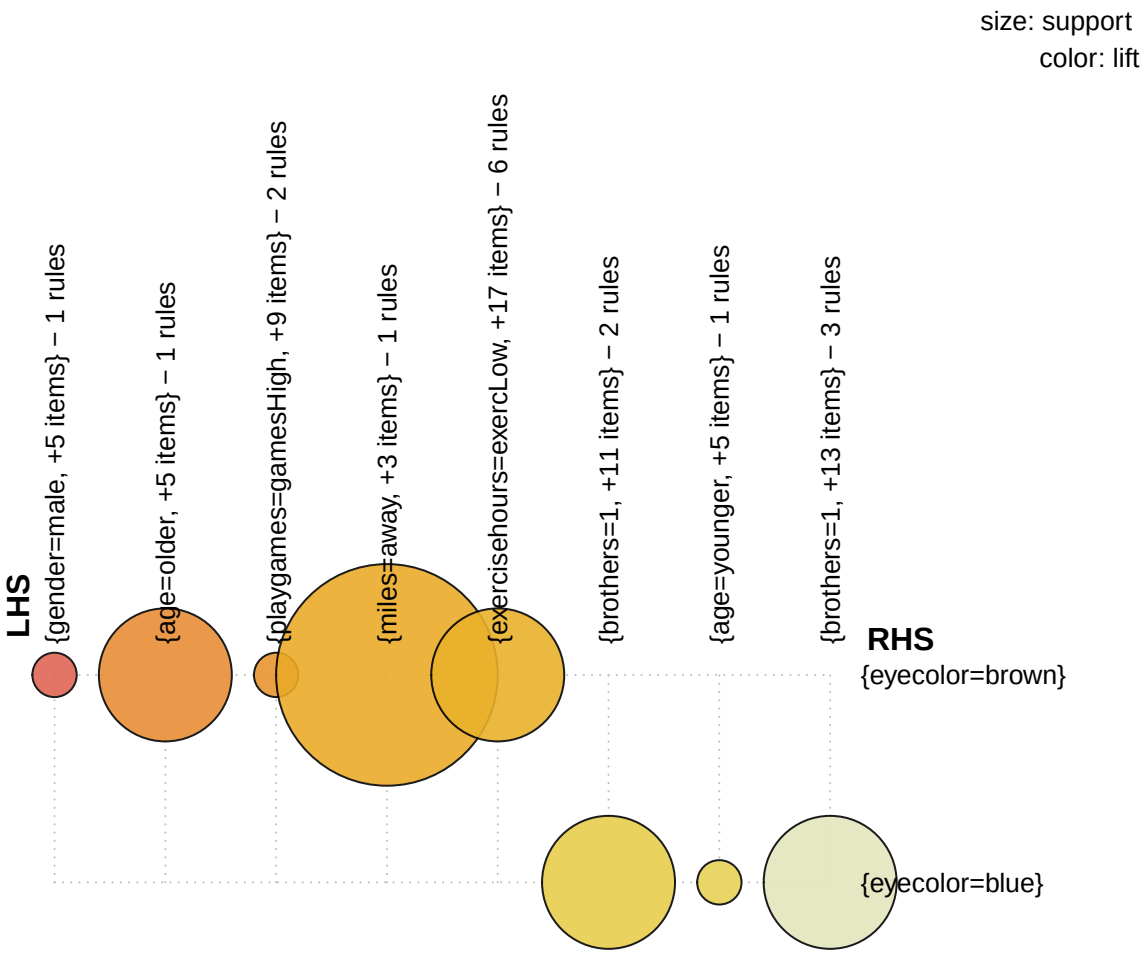
```
## 9  {height=taller,
##     miles=closer,
##     brothers=0,
##     sisters=1,
##     exercisehours=exercLow,
##     musiccds=musicLow}       => {eyecolor=brown} 0.005802708  0.8000000 2.576947
## 10 {gender=female,
##     age=older,
##     brothers=1,
##     computertime=compLow,
##     exercise=Yes,
##     exercisehours=exercLow}  => {eyecolor=brown} 0.005802708  0.8000000 2.576947
## 11 {age=older,
##     height=taller,
##     miles=away,
##     computertime=compLow,
##     exercisehours=exercLow,
##     playgames=gamesHigh}     => {eyecolor=brown} 0.005802708  0.8000000 2.576947
## 12 {miles=away,
##     brothers=1,
##     sisters=1,
##     computertime=compLow,
##     exercisehours=exercHigh,
##     musiccds=musicLow,
##     watchtv=TVLow}           => {eyecolor=blue}  0.005802708  0.8571429 2.431511
## 13 {gender=male,
##     age=older,
##     miles=closer,
##     brothers=1,
##     sisters=1,
##     computertime=compLow,
##     playgames=gamesHigh,
##     watchtv=TVHigh}          => {eyecolor=blue}  0.005802708  0.8571429 2.431511
## 14 {age=younger,
##     year=first,
##     sisters=1,
##     exercisehours=exercHigh,
##     musiccds=musicLow,
##     watchtv=TVLow}           => {eyecolor=blue}  0.005319149  0.8461538 2.400338
## 15 {gender=male,
##     height=shorter,
##     brothers=1,
##     exercisehours=exercLow,
##     playgames=gamesLow,
##     watchtv=TVHigh}          => {eyecolor=blue}  0.005802708  0.8000000 2.269410
## 16 {gender=male,
##     year=third,
##     brothers=1,
##     sisters=1,
##     computertime=compLow,
##     exercise=Yes,
##     musiccds=musicLow}       => {eyecolor=blue}  0.005802708  0.8000000 2.269410
## 17 {age=older,
##     miles=closer,
```

```
##     brothers=1,
##     sisters=1,
##     computertime=compLow,
##     playgames=gamesHigh,
##     watchtv=TVHigh}          => {eyecolor=blue}  0.005802708  0.8000000 2.269410
```

Results

The 17 remaining rules are shown graphically in the figure. It is balloon plot with antecedent groups (lhs) as columns and consequents as rows (rhs). The colour of the balloons represent the aggregated lift in the group with a certain consequent and the size of the balloon shows the aggregated support. The number of antecedents and the most important (frequent) items in the group are displayed as the labels for the columns. Furthermore, the columns and rows in the plot are reordered such that the lift is decreasing from top down and from left to right, placing the most interesting group in the top left corner. It is the rule which contain "males"" and 5 other items (sisters=0, computertime=compHigh, exercise=No, musiccds=musicLow, playgames=gamesLow) in the antecedent and the consequent is "brown eyecolour". The support value for this rule is 0.005 and means the antecedents and the consequent appeared together in 0.5% of the cases. The confidence of 0.92 for this rule implies that when a student has the previously specified antecedents, 92% of the time he has brown eyes. To be noted that the other 3 categories of eye colour are not represented by a rule.

```
plot(rules.pruned, method="grouped")
```



Discussion

Association Rules identified items with high confidence. Nevertheless, 3 out of the 5 eye colour categories where not represented in the results. This can be interpreted as absence of association or correlation between groups of variables and the 3 remaining categories. It can be attributed or to a real absence of association or a limitation of the algorithm. Such a limitation could be the restrictive form of the data to which it can be applied, namely the discretisation of continuous variables. Another limitation is that rules with high confidence or lift, but low support, are not discovered. To overcome this shortcoming we fixed a low support value which also made sense for our dataset.

References

Agrawal, Rakesh, Ramakrishnan Srikant, and others. 1994. "Fast Algorithms for Mining Association Rules." In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, 1215:487–99.

Froelich, Amy G, and W Robert Stephenson. 2013. "Does Eye Color Depend on Gender? It Might Depend

on Who or How You Ask." *Journal of Statistics Education* 21 (2).