Association Rule Mining with R*

Yanchang Zhao
http://www.RDataMining.com

R and Data Mining Workshop
for the Master of Business Analytics course, Deakin University, Melbourne

28 May 2015

*Presented at Australian Customs (Canberra, Australia) in Oct 2014, at AusDM 2014 (QUT, Brisbane) in Nov 2014, at Twitter (US) in Oct 2014, at UJAT (Mexico) in Sept 2014, and at University of Canberra in Sept 2013

Outline

Association Rule Mining with R [†]

- basic concepts of association rules
- association rules mining with R
- pruning redundant rules
- interpreting and visualizing association rules
- recommended readings

[†] Chapter 9: Association Rules, *R and Data Mining: Examples and Case Studies*. http://www.rdatamining.com/docs/RDataMining.pdf

Association Rules

Association rules are rules presenting association or correlation between itemsets.

$$
\begin{aligned}
\text{support}(A \Rightarrow B) &= P(A \cup B) \\
\text{confidence}(A \Rightarrow B) &= P(B|A) \\
&= \frac{P(A \cup B)}{P(A)} \\
\text{lift}(A \Rightarrow B) &= \frac{\text{confidence}(A \Rightarrow B)}{P(B)} \\
&= \frac{P(A \cup B)}{P(A)P(B)}
\end{aligned}
$$

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

Association Rule Mining Algorithms in R

- APRIORI
  - a level-wise, breadth-first algorithm which counts transactions to find frequent itemsets and then derive association rules from them
  - apriori() in package arules
- ECLAT
  - finds frequent itemsets with equivalence classes, depth-first search and set intersection instead of counting
  - eclat() in the same package

Outline

The Titanic Dataset

- The Titanic dataset in the *datasets* package is a 4-dimensional table with summarized information on the fate of passengers on the Titanic according to social class, sex, age and survival.
- To make it suitable for association rule mining, we reconstruct the raw data as `titanic.raw`, where each row represents a person.
- The reconstructed raw data can also be downloaded at `http://www.rdatamining.com/data/titanic.raw.rdata`.

```r
load("./data/titanic.raw.rdata")
## draw a sample of 5 records
idx <- sample(1:nrow(titanic.raw), 5)
titanic.raw[idx, ]
```

```
##       Class  Sex   Age Survived
## 1203   Crew Male Adult       No
## 1218   Crew Male Adult       No
## 1674    3rd Male Adult      Yes
## 941    Crew Male Adult       No
## 820    Crew Male Adult       No
```

```r
summary(titanic.raw)
```

```
##   Class         Sex          Age        Survived
##  1st :325   Female: 470   Adult:2092   No :1490
##  2nd :285   Male  :1731   Child: 109   Yes: 711
##  3rd :706
##  Crew:885
```

Function apriori()

Mine frequent itemsets, association rules or association hyperedges using the Apriori algorithm. The Apriori algorithm employs level-wise search for frequent itemsets.

Default settings:

- minimum support: supp=0.1
- minimum confidence: conf=0.8
- maximum length of rules: maxlen=10

```r
library(arules)
rules.all <- apriori(titanic.raw)

##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport support
##         0.8    0.1    1 none FALSE            TRUE     0.1
##  minlen maxlen target   ext
##       1     10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## apriori - find association rules with the apriori algorithm
## version 4.21 (2004.05.09)        (c) 1996-2004   Christian ...
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[10 item(s), 2201 transaction(s)] done ...
## sorting and recoding items ... [9 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [27 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
inspect(rules.all)

##    lhs               rhs            support   confidence    ...
## 1  {}             => {Age=Adult}    0.9504771  0.9504771 1.0...
## 2  {Class=2nd}    => {Age=Adult}    0.1185825  0.9157895 0.9...
## 3  {Class=1st}    => {Age=Adult}    0.1449341  0.9815385 1.0...
## 4  {Sex=Female}   => {Age=Adult}    0.1930940  0.9042553 0.9...
## 5  {Class=3rd}    => {Age=Adult}    0.2848705  0.8881020 0.9...
## 6  {Survived=Yes} => {Age=Adult}    0.2971377  0.9198312 0.9...
## 7  {Class=Crew}   => {Sex=Male}     0.3916402  0.9740113 1.2...
## 8  {Class=Crew}   => {Age=Adult}    0.4020900  1.0000000 1.0...
## 9  {Survived=No}  => {Sex=Male}     0.6197183  0.9154362 1.1...
## 10 {Survived=No}  => {Age=Adult}    0.6533394  0.9651007 1.0...
## 11 {Sex=Male}     => {Age=Adult}    0.7573830  0.9630272 1.0...
## 12 {Sex=Female,                                             ...
##     Survived=Yes} => {Age=Adult}    0.1435711  0.9186047 0.9...
## 13 {Class=3rd,                                              ...
##     Sex=Male}     => {Survived=No} 0.1917310  0.8274510 1.2...
## 14 {Class=3rd,                                              ...
##     Survived=No}  => {Age=Adult}    0.2162653  0.9015152 0.9...
## 15 {Class=3rd,                                              ...
##     Sex=Male}     => {Age=Adult}    0.2099046  0.9058824 0.9...
## 16 {Sex=Male,                                               ...
##     Survived=Yes} => {Age=Adult}    0.1535666  0.9209809 0.9...
```

```r
# rules with rhs containing "Survived" only
rules <- apriori(titanic.raw,
                 control = list(verbose=F),
                 parameter = list(minlen=2, supp=0.005, conf=0.8),
                 appearance = list(rhs=c("Survived=No",
                                         "Survived=Yes"),
                                   default="lhs"))
## keep three decimal places
quality(rules) <- round(quality(rules), digits=3)
## order rules by lift
rules.sorted <- sort(rules, by="lift")
```

```r
inspect(rules.sorted)

##    lhs               rhs            support confidence  lift
## 1  {Class=2nd,
##     Age=Child}   => {Survived=Yes}   0.011      1.000 3.096
## 2  {Class=2nd,
##     Sex=Female,
##     Age=Child}   => {Survived=Yes}   0.006      1.000 3.096
## 3  {Class=1st,
##     Sex=Female} => {Survived=Yes}   0.064      0.972 3.010
## 4  {Class=1st,
##     Sex=Female,
##     Age=Adult}   => {Survived=Yes}   0.064      0.972 3.010
## 5  {Class=2nd,
##     Sex=Female} => {Survived=Yes}   0.042      0.877 2.716
## 6  {Class=Crew,
##     Sex=Female} => {Survived=Yes}   0.009      0.870 2.692
## 7  {Class=Crew,
##     Sex=Female,
##     Age=Adult}   => {Survived=Yes}   0.009      0.870 2.692
## 8  {Class=2nd,
##     Sex=Female,
##     Age=Adult}   => {Survived=Yes}   0.036      0.860 2.663
## 9  {Class=2nd,
```

Outline

Redundant Rules

```
inspect(rules.sorted[1:2])

##   lhs                rhs              support confidence  lift
## 1 {Class=2nd,
##    Age=Child}  => {Survived=Yes}   0.011          1 3.096
## 2 {Class=2nd,
##    Sex=Female,
##    Age=Child}  => {Survived=Yes}   0.006          1 3.096
```

- Rule #2 provides no extra knowledge in addition to rule #1, since rules #1 tells us that all 2nd-class children survived.
- When a rule (such as #2) is a super rule of another rule (#1) and the former has the same or a lower lift, the former rule (#2) is considered to be redundant.
- Other redundant rules in the above result are rules #4, #7 and #8, compared respectively with #3, #6 and #5.

Remove Redundant Rules

```r
## find redundant rules
subset.matrix <- is.subset(rules.sorted, rules.sorted)
subset.matrix[lower.tri(subset.matrix, diag = T)] <- NA
redundant <- colSums(subset.matrix, na.rm = T) >= 1
```

```r
## which rules are redundant
which(redundant)

## [1] 2 4 7 8

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

Remaining Rules

```
inspect(rules.pruned)

##   lhs                rhs             support confidence  lift
## 1 {Class=2nd,
##    Age=Child}   => {Survived=Yes}   0.011        1.000 3.096
## 2 {Class=1st,
##    Sex=Female} => {Survived=Yes}   0.064        0.972 3.010
## 3 {Class=2nd,
##    Sex=Female} => {Survived=Yes}   0.042        0.877 2.716
## 4 {Class=Crew,
##    Sex=Female} => {Survived=Yes}   0.009        0.870 2.692
## 5 {Class=2nd,
##    Sex=Male,
##    Age=Adult}  => {Survived=No}    0.070        0.917 1.354
## 6 {Class=2nd,
##    Sex=Male}   => {Survived=No}    0.070        0.860 1.271
## 7 {Class=3rd,
##    Sex=Male,
##    Age=Adult}  => {Survived=No}    0.176        0.838 1.237
## 8 {Class=3rd,
##    Sex=Male}   => {Survived=No}    0.192        0.827 1.222
```

Outline

```r
inspect(rules.pruned[1])
```

```
##   lhs               rhs            support confidence  lift
## 1 {Class=2nd,
##    Age=Child} => {Survived=Yes}   0.011          1 3.096
```

Did children of the 2nd class have a higher survival rate than other children?

```r
inspect(rules.pruned[1])

##   lhs               rhs            support confidence  lift
## 1 {Class=2nd,
##    Age=Child} => {Survived=Yes}   0.011          1 3.096
```

Did children of the 2nd class have a higher survival rate than other children?

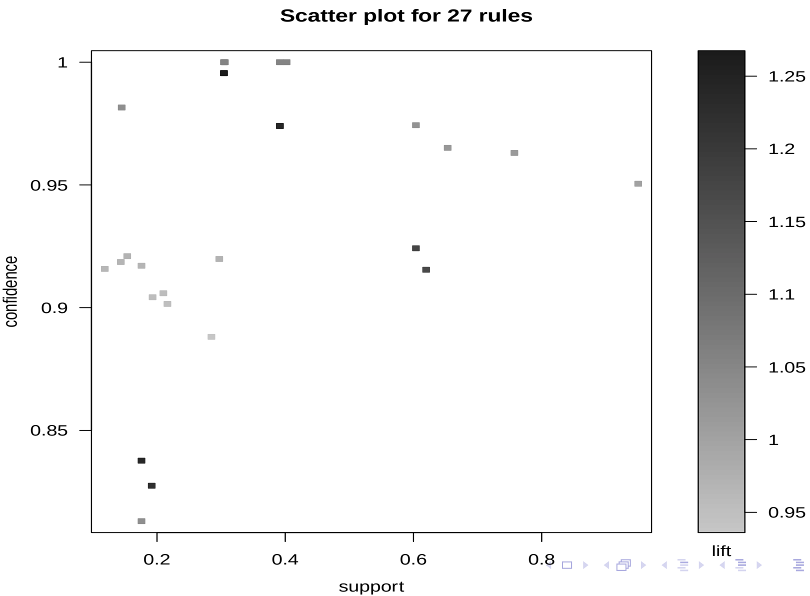
The rule states only that all children of class 2 survived, but provides no information at all to compare the survival rates of different classes.

Rules about Children

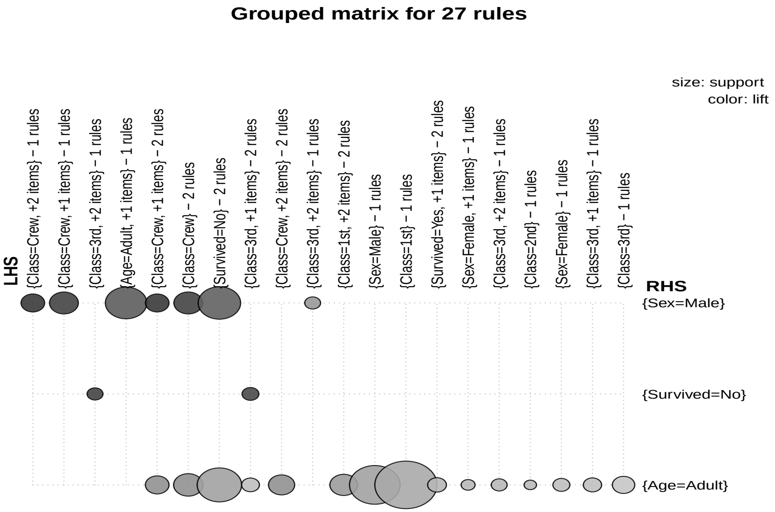
```
rules <- apriori(titanic.raw, control = list(verbose=F),
     parameter = list(minlen=3, supp=0.002, conf=0.2),
     appearance = list(default="none", rhs=c("Survived=Yes"),
                       lhs=c("Class=1st", "Class=2nd", "Class=3rd",
                             "Age=Child", "Age=Adult")))
rules.sorted <- sort(rules, by="confidence")
inspect(rules.sorted)

##   lhs              rhs                support confidence     ...
## 1 {Class=2nd,                                                ...
##    Age=Child} => {Survived=Yes} 0.010904134  1.0000000 3.09...
## 2 {Class=1st,                                                ...
##    Age=Child} => {Survived=Yes} 0.002726034  1.0000000 3.09...
## 3 {Class=1st,                                                ...
##    Age=Adult} => {Survived=Yes} 0.089504771  0.6175549 1.91...
## 4 {Class=2nd,                                                ...
##    Age=Adult} => {Survived=Yes} 0.042707860  0.3601533 1.11...
## 5 {Class=3rd,                                                ...
##    Age=Child} => {Survived=Yes} 0.012267151  0.3417722 1.05...
## 6 {Class=3rd,                                                ...
##    Age=Adult} => {Survived=Yes} 0.068605179  0.2408293 0.74...
```

Outline

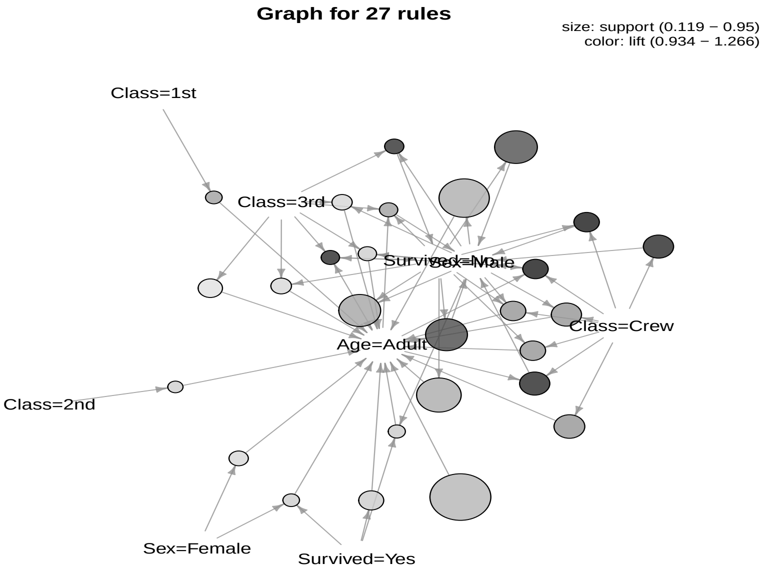
```r
library(arulesViz)
plot(rules.all)
```



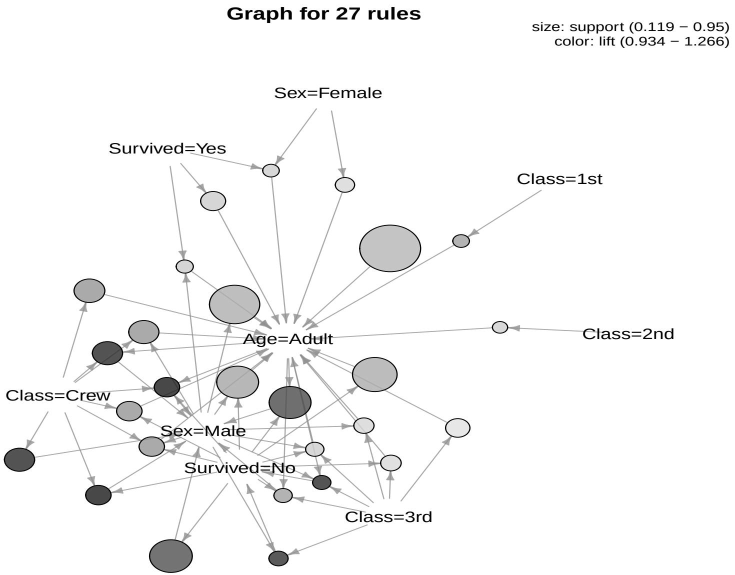
```r
plot(rules.all, method = "grouped")
```



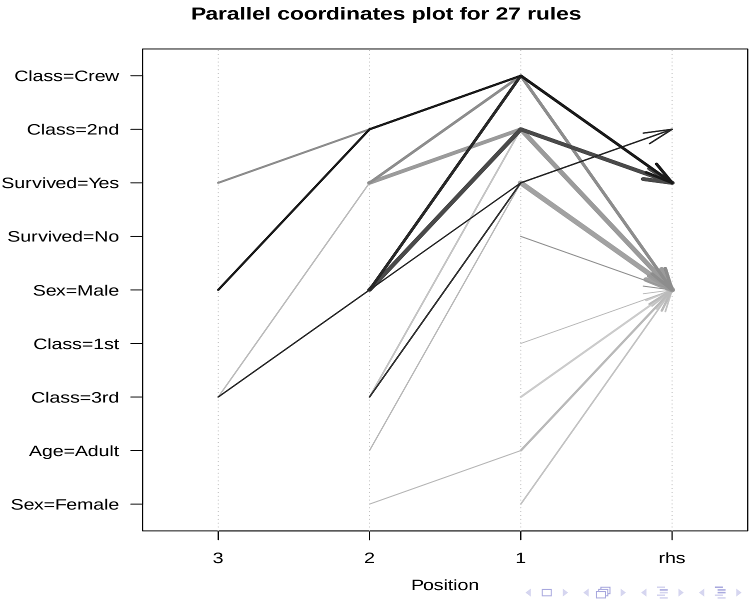
```r
plot(rules.all, method = "graph")
```



```
plot(rules.all, method = "graph", control = list(type = "items"))
```



```r
plot(rules.all, method = "paracoord", control = list(reorder = TRUE))
```



Outline

Further Readings

- More than 20 interestingness measures, such as chi-square, conviction, gini and leverage

  Tan, P.-N., Kumar, V., and Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. In Proc. of KDD '02, pages 32-41, New York, NY, USA. ACM Press.
- Post mining of association rules, such as selecting interesting association rules, visualization of association rules and using association rules for classification

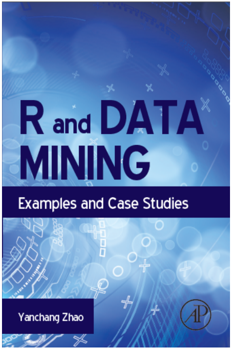
  Yanchang Zhao, Chengqi Zhang and Longbing Cao (Eds.). "Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction", ISBN 978-1-60566-404-0, May 2009. Information Science Reference.
- Package *arulesSequences*: mining sequential patterns

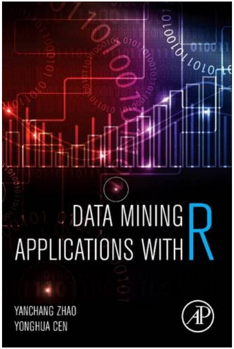
  `http://cran.r-project.org/web/packages/arulesSequences/`

Online Resources

- Chapter 9: Association Rules, in book *R and Data Mining: Examples and Case Studies*
  `http://www.rdatamining.com/docs/RDataMining.pdf`
- R Reference Card for Data Mining
  `http://www.rdatamining.com/docs/R-refcard-data-mining.pdf`
- Free online courses and documents
  `http://www.rdatamining.com/resources/`
- RDataMining Group on LinkedIn (12,000+ members)
  `http://group.rdatamining.com`
- RDataMining on Twitter (2,000+ followers)
  `@RDataMining`

The End





Thanks!

Email: `yanchang(at)rdatamining.com`