Association Rule Mining with R *

Yanchang Zhao

http://www.RDataMining.com

Machine Learning 102 Workshop
S P Jain School of Global Management, Mumbai, India

30 May - 5 June 2016

*Chapter 9 - Association Rules, in *R and Data Mining: Examples and Case Studies*. http://www.rdatamining.com/docs/RDataMining-book.pdf

Outline

Association Rules

Association rules are rules presenting association or correlation between itemsets.

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

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

Association Rule Mining Algorithms in R

- Apriori [Agrawal and Srikant, 1994]
  - 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 [Zaki et al., 1997]
  - finds frequent itemsets with equivalence classes, depth-first search and set intersection instead of counting
  - `eclat()` in package *arules*

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
## 314    2nd   Male Adult       No
## 1916   1st Female Adult      Yes
## 1511   3rd   Male Child      Yes
## 1408   3rd Female Adult       No
## 6      3rd   Male Child       No

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   Christia...
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[10 item(s), 2201 transaction(s)] don...
## 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...
## 2  {Class=2nd}     => {Age=Adult}    0.1185825 0.9157895 0...
## 3  {Class=1st}     => {Age=Adult}    0.1449341 0.9815385 1...
## 4  {Sex=Female}    => {Age=Adult}    0.1930940 0.9042553 0...
## 5  {Class=3rd}     => {Age=Adult}    0.2848705 0.8881020 0...
## 6  {Survived=Yes} => {Age=Adult}    0.2971377 0.9198312 0...
## 7  {Class=Crew}    => {Sex=Male}     0.3916402 0.9740113 1...
## 8  {Class=Crew}    => {Age=Adult}    0.4020900 1.0000000 1...
## 9  {Survived=No}   => {Sex=Male}     0.6197183 0.9154362 1...
## 10 {Survived=No}   => {Age=Adult}    0.6533394 0.9651007 1...
## 11 {Sex=Male}      => {Age=Adult}    0.7573830 0.9630272 1...
## 12 {Sex=Female,                                          ...
##     Survived=Yes} => {Age=Adult}    0.1435711 0.9186047 0...
## 13 {Class=3rd,                                           ...
##     Sex=Male}      => {Survived=No} 0.1917310 0.8274510 1...
## 14 {Class=3rd,                                           ...
##     Survived=No}   => {Age=Adult}    0.2162653 0.9015152 0...
## 15 {Class=3rd,                                           ...
##     Sex=Male}      => {Age=Adult}    0.2099046 0.9058824 0...
## 16 {Sex=Male,                                            ...
##     Survived=Yes} => {Age=Adult}    0.1535666 0.9209809 0...
```

```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

- There are often too many association rules discovered from a dataset.
- It is necessary to remove redundant rules before a user is able to study the rules and identify interesting ones from them.

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

```
## 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
```

```
## 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.

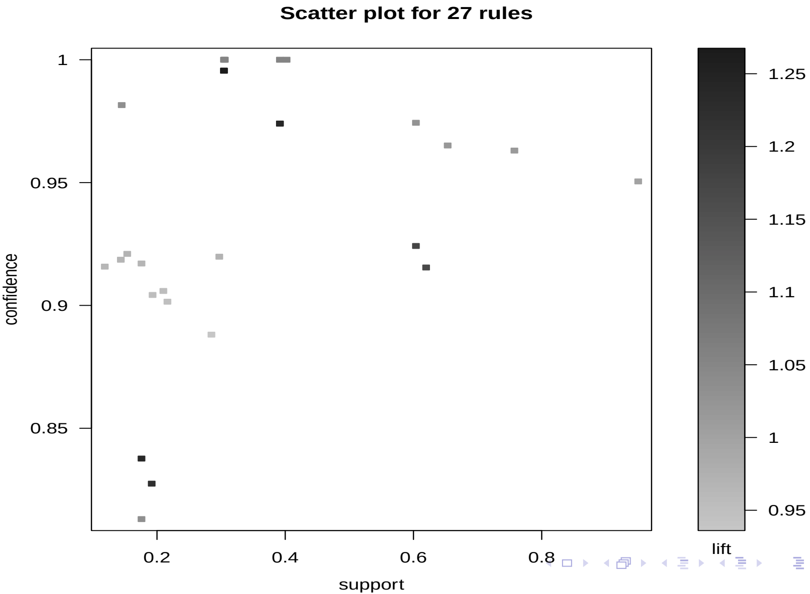
Rules about Children

```
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)
```

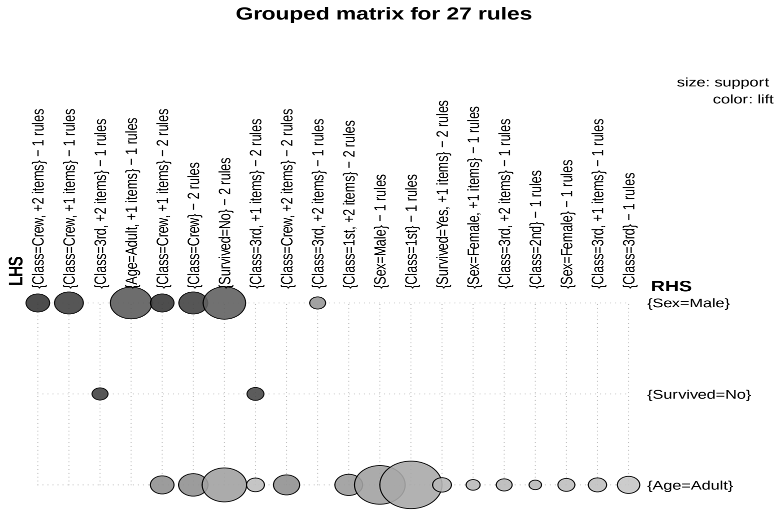
```
##   lhs              rhs                support confidence   ...
## 1 {Class=2nd,                                              ...
##    Age=Child} => {Survived=Yes} 0.010904134  1.0000000 3....
## 2 {Class=1st,                                              ...
##    Age=Child} => {Survived=Yes} 0.002726034  1.0000000 3....
## 3 {Class=1st,                                              ...
##    Age=Adult} => {Survived=Yes} 0.089504771  0.6175549 1....
## 4 {Class=2nd,                                              ...
##    Age=Adult} => {Survived=Yes} 0.042707860  0.3601533 1....
## 5 {Class=3rd,                                              ...
##    Age=Child} => {Survived=Yes} 0.012267151  0.3417722 1....
## 6 {Class=3rd,                                              ...
##    Age=Adult} => {Survived=Yes} 0.068605179  0.2408293 0....
```

Outline

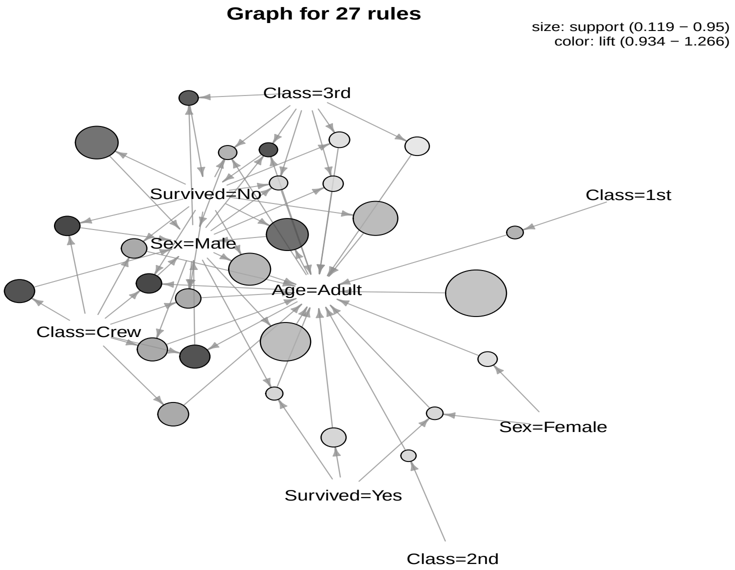
```r
library(arulesViz)
plot(rules.all)
```



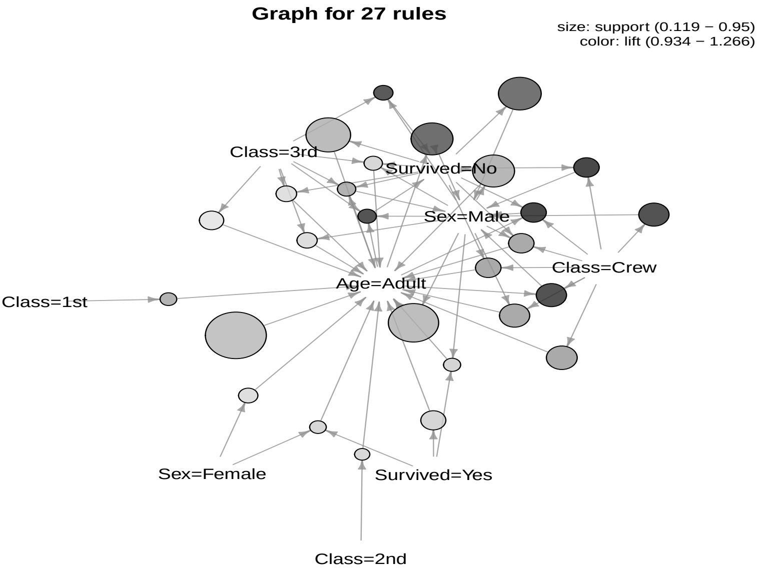
```r
plot(rules.all, method = "grouped")
```



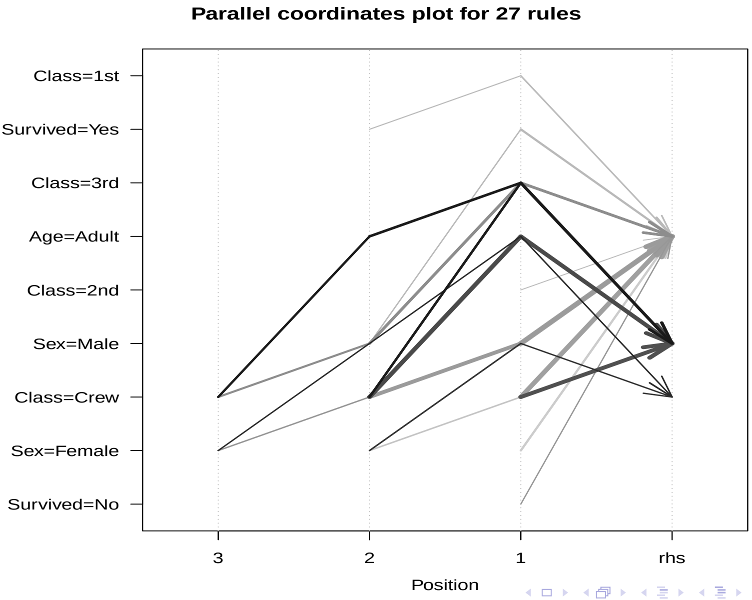
```r
plot(rules.all, method = "graph")
```



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



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



Outline

Further Readings

- Data Mining Algorithms In R: Apriori
  https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_Apriori_Algorithm
- Data Mining Algorithms In R: ECLAT
  https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_Eclat_Algorithm
- Data Mining Algorithms In R: FP-Growth
  https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_FP-Growth_Algorithm
- FP-Growth Implementation by Christian Borgelt
  http://www.borgelt.net/fpgrowth.html
- Frequent Itemset Mining Implementations Repository
  http://fimi.ua.ac.be/data/
- 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 [Zhao, 2012]
  `http://www.rdatamining.com/docs/RDataMining-book.pdf`
- RDataMining Reference Card
  `http://www.rdatamining.com/docs/RDataMining-reference-card.pdf`
- Free online courses and documents
  `http://www.rdatamining.com/resources/`
- RDataMining Group on LinkedIn (20,000+ members)
  `http://group.rdatamining.com`
- Twitter (2,500+ followers)
  @RDataMining

Outline

The Mushroom Dataset I

- The mushroom dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms †.
- A csv file with 8,124 observations on 23 categorical variables:
  1. class: edible=e, poisonous=p
  2. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
  3. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
  4. cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r, pink=p,purple=u,red=e,white=w,yellow=y
  5. bruises?: bruises=t,no=f
  6. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s
  7. gill-attachment: attached=a,descending=d,free=f,notched=n
  8. gill-spacing: close=c,crowded=w,distant=d
  9. gill-size: broad=b,narrow=n
  10. gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e, white=w,yellow=y

The Mushroom Dataset II

11. stalk-shape: enlarging=e,tapering=t
12. stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?
13. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
14. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
15. stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
16. stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
17. veil-type: partial=p,universal=u
18. veil-color: brown=n,orange=o,white=w,yellow=y
19. ring-number: none=n,one=o,two=t
20. ring-type: cobwebby=c,evanescent=e,flaring=f,large=l, none=n,pendant=p,sheathing=s,zone=z

The Mushroom Dataset III

21. spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r, orange=o,purple=u,white=w,yellow=y
22. population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y
23. habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

† `https://archive.ics.uci.edu/ml/datasets/Mushroom`

Load Mushroom Dataset

```r
## load mushroom data from UCI the Machine Learning Repository
url <- paste0("http://archive.ics.uci.edu/ml/",
    "machine-learning-databases/mushroom/agaricus-lepiota.data")
```

```r
mushrooms <- read.csv(file = url, header = FALSE)
names(mushrooms) <- c("class", "cap-shape", "cap-surface",
    "cap-color", "bruises", "odor", "gill-attachment", "gill-spacing",
    "gill-size", "gill-color", "stalk-shape", "stalk-root",
    "stalk-surface-above-ring", "stalk-surface-below-ring",
    "stalk-color-above-ring", "stalk-color-below-ring",
    "veil-type", "veil-color", "ring-number", "ring-type",
    "spore-print-color", "population", "habitat")
table(mushrooms$class, useNA="ifany")

##
##    e    p
## 4208 3916
```

The Mushroom Dataset

```
str(mushrooms)

## 'data.frame': 8124 obs. of  23 variables:
##  $ class                   : Factor w/ 2 levels "e","p": ...
##  $ cap-shape               : Factor w/ 6 levels "b","c","...
##  $ cap-surface             : Factor w/ 4 levels "f","g","...
##  $ cap-color               : Factor w/ 10 levels "b","c",...
##  $ bruises                 : Factor w/ 2 levels "f","t": ...
##  $ odor                    : Factor w/ 9 levels "a","c","...
##  $ gill-attachment         : Factor w/ 2 levels "a","f": ...
##  $ gill-spacing            : Factor w/ 2 levels "c","w": ...
##  $ gill-size               : Factor w/ 2 levels "b","n": ...
##  $ gill-color              : Factor w/ 12 levels "b","e",...
##  $ stalk-shape             : Factor w/ 2 levels "e","t": ...
##  $ stalk-root              : Factor w/ 5 levels "?","b","...
##  $ stalk-surface-above-ring: Factor w/ 4 levels "f","k","...
##  $ stalk-surface-below-ring: Factor w/ 4 levels "f","k","...
##  $ stalk-color-above-ring  : Factor w/ 9 levels "b","c","...
##  $ stalk-color-below-ring  : Factor w/ 9 levels "b","c","...
##  $ veil-type               : Factor w/ 1 level "p": 1 1 1...
##  $ veil-color              : Factor w/ 4 levels "n","o","...
##  $ ring-number             : Factor w/ 3 levels "n","o","...
```

Exercise

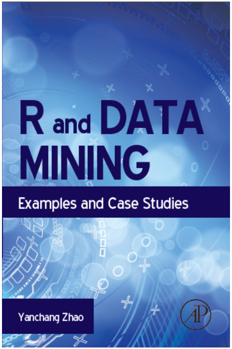
- From the mushroom data, find association rules that can be used to identify the edibility of a mushroom
- Think about parameters: length of rules, minimum support, minimum confidence
- How to find only rules relevant to edibility?
- Which interestingness measures to use?
- Any reduntant rules? How to remove them?
- What are characteristics of edible mushrooms? And characteristics of poisonous ones?

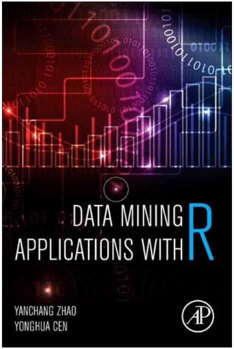
Mining Association Rules from Mushroom Dataset

```
rules <- apriori(mushrooms, control = list(verbose=F),
                 parameter = list(minlen=2, maxlen=5),
                 appearance = list(rhs=c("class=p", "class=e"),
                                   default="lhs"))
quality(rules) <- round(quality(rules), digits=3)
rules.sorted <- sort(rules, by="confidence")
inspect(head(rules.sorted))
```

```
##   lhs                 rhs       support confidence  lift
## 1 {ring-type=l}    => {class=p}   0.160          1 2.075
## 2 {gill-color=b}   => {class=p}   0.213          1 2.075
## 3 {odor=f}         => {class=p}   0.266          1 2.075
## 4 {gill-size=b,
##    gill-color=n}   => {class=e}   0.108          1 1.931
## 5 {odor=n,
##    stalk-root=e}   => {class=e}   0.106          1 1.931
## 6 {bruises=f,
##    stalk-root=e}   => {class=e}   0.106          1 1.931
```

The End





Thanks!

Email: yanchang(at)rdatamining.com

References

Agrawal, R. and Srikant, R. (1994).
Fast algorithms for mining association rules in large databases.
In *Proc. of the 20th International Conference on Very Large Data Bases*, pages 487–499, Santiago, Chile.

Zaki, M. J., Parthasarathy, S., Ogihara, M., and Li, W. (1997).
New algorithms for fast discovery of association rules.
Technical Report 651, Computer Science Department, University of Rochester, Rochester, NY 14627.

Zhao, Y. (2012).
R and Data Mining: Examples and Case Studies.
Academic Press, Elsevier.