

# Association Rule Mining with R\*

Yanchang Zhao

<http://www.RDataMining.com>

30 October 2014

---

\*Presented at Australian Customs, Amazon and UJAT

# Outline

Introduction

Association Rule Mining

Removing Redundancy

Interpreting Rules

Visualizing Association Rules

Further Readings and Online Resources

# Association Rule Mining with R <sup>†</sup>

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

---

<sup>†</sup>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

Introduction

Association Rule Mining

Removing Redundancy

Interpreting Rules

Visualizing Association Rules

Further Readings and Online Resources

# 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>.

```
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
## 2096   2nd Female Adult      Yes
## 1210   Crew   Male Adult      No
## 329    3rd   Male Adult      No
## 1727   Crew   Male Adult      Yes
## 1305   Crew   Male Adult      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`

```

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].

```

```
inspect(rules.all)
```

##	lhs	rhs	support	confidence	lift
## 1	{}	=> {Age=Adult}	0.9505	0.9505	1.0000
## 2	{Class=2nd}	=> {Age=Adult}	0.1186	0.9158	0.9635
## 3	{Class=1st}	=> {Age=Adult}	0.1449	0.9815	1.0327
## 4	{Sex=Female}	=> {Age=Adult}	0.1931	0.9043	0.9514
## 5	{Class=3rd}	=> {Age=Adult}	0.2849	0.8881	0.9344
## 6	{Survived=Yes}	=> {Age=Adult}	0.2971	0.9198	0.9678
## 7	{Class=Crew}	=> {Sex=Male}	0.3916	0.9740	1.2385
## 8	{Class=Crew}	=> {Age=Adult}	0.4021	1.0000	1.0521
## 9	{Survived=No}	=> {Sex=Male}	0.6197	0.9154	1.1640
## 10	{Survived=No}	=> {Age=Adult}	0.6533	0.9651	1.0154
## 11	{Sex=Male}	=> {Age=Adult}	0.7574	0.9630	1.0132
## 12	{Sex=Female, Survived=Yes}	=> {Age=Adult}	0.1436	0.9186	0.9665
## 13	{Class=3rd, Sex=Male}	=> {Survived=No}	0.1917	0.8275	1.2223
## 14	{Class=3rd, Survived=No}	=> {Age=Adult}	0.2163	0.9015	0.9485
## 15	{Class=3rd, Sex=Male}	=> {Age=Adult}	0.2099	0.9059	0.9531
## 16	{Sex=Male, Survived=Yes}	=> {Age=Adult}	0.1536	0.9210	0.9690

```
# 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")
```

```
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

Introduction

Association Rule Mining

Removing Redundancy

Interpreting Rules

Visualizing Association Rules

Further Readings and Online Resources

# 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 rule #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

Introduction

Association Rule Mining

Removing Redundancy

**Interpreting Rules**

Visualizing Association Rules

Further Readings and Online Resources

```
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?

```
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	lift
## 1	{Class=2nd,				
##	Age=Child}	=> {Survived=Yes}	0.010904	1.0000	3.0956
## 2	{Class=1st,				
##	Age=Child}	=> {Survived=Yes}	0.002726	1.0000	3.0956
## 3	{Class=1st,				
##	Age=Adult}	=> {Survived=Yes}	0.089505	0.6176	1.9117
## 4	{Class=2nd,				
##	Age=Adult}	=> {Survived=Yes}	0.042708	0.3602	1.1149
## 5	{Class=3rd,				
##	Age=Child}	=> {Survived=Yes}	0.012267	0.3418	1.0580
## 6	{Class=3rd,				
##	Age=Adult}	=> {Survived=Yes}	0.068605	0.2408	0.7455

# Outline

Introduction

Association Rule Mining

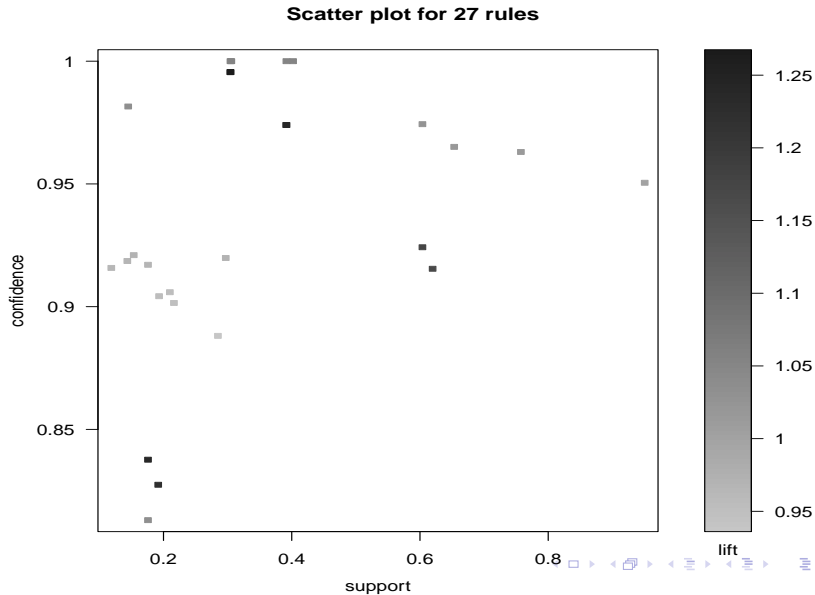
Removing Redundancy

Interpreting Rules

**Visualizing Association Rules**

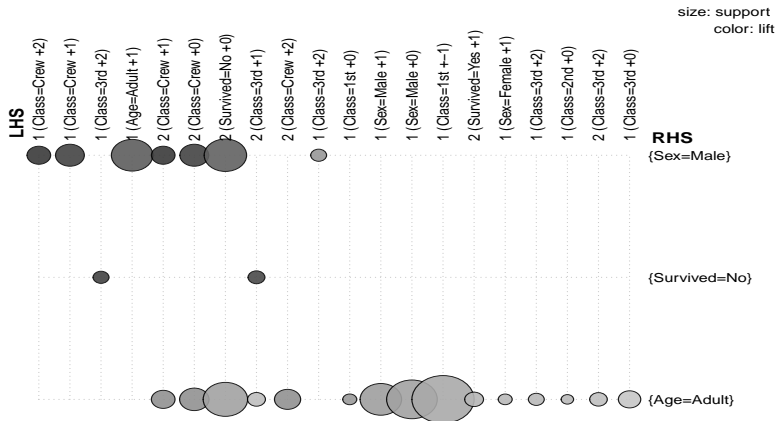
Further Readings and Online Resources

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



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

## Grouped matrix for 27 rules



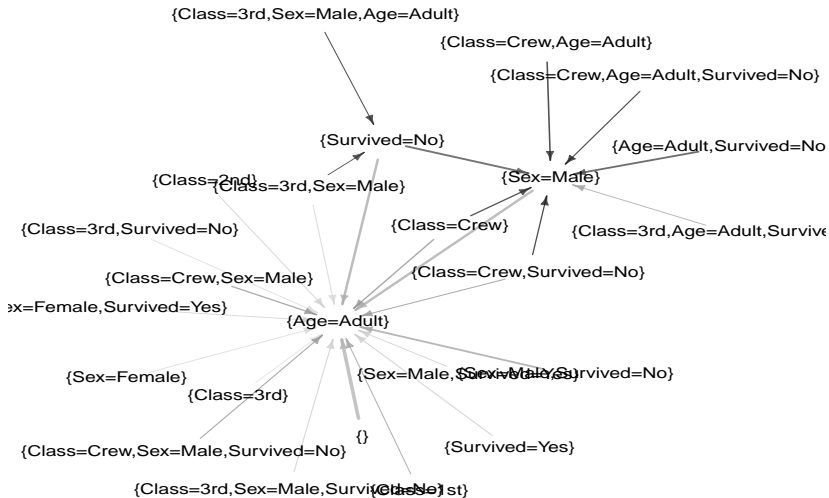


```
plot(rules.all, method = "graph")
```

### Graph for 27 rules

width: support (0.119 – 0.95)

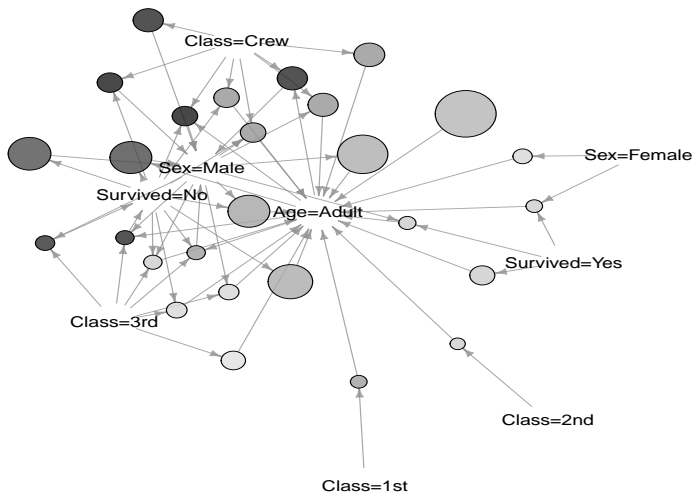
color: lift (0.934 – 1.266)



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

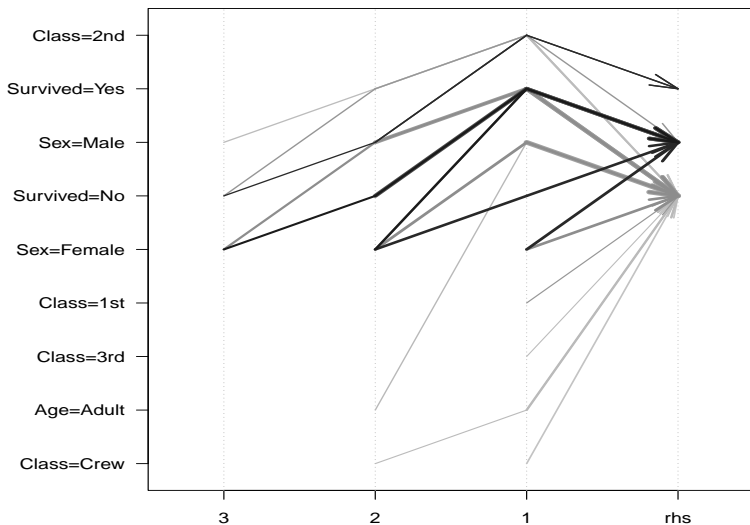
**Graph for 27 rules**

size: support (0.119 – 0.95)  
color: lift (0.934 – 1.266)



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

**Parallel coordinates plot for 27 rules**



Position

# Outline

Introduction

Association Rule Mining

Removing Redundancy

Interpreting Rules

Visualizing Association Rules

Further Readings and Online Resources

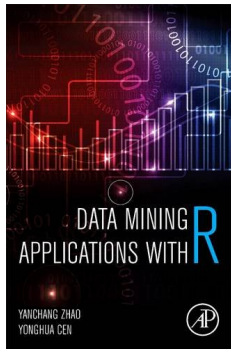
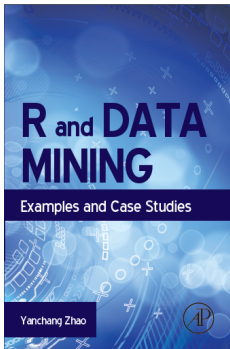
## Further Readings

- ▶ 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 (7,000+ members)  
<http://group.rdatamining.com>
- ▶ RDataMining on Twitter (1,800+ followers)  
[@RDataMining](#)

# The End



Thanks!

Email: [yanchang\(at\)rdatamining.com](mailto:yanchang(at)rdatamining.com)