Association Rule Mining with R *

Yanchang Zhao

http://www.RDataMining.com

Short Course on R and Data Mining

University of Canberra

7 October 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

- To discover association rules showing itemsets that occur together frequently [Agrawal et al., 1993].
- Widely used to analyze retail basket or transaction data.
- An association rule is of the form $A \Rightarrow B$, where A and B are items or attribute-value pairs.
- The rule means that those database tuples having the items in the left hand of the rule are also likely to having those items in the right hand.
- Examples of association rules:
  - *bread* $\Rightarrow$ *butter*
  - *computer* $\Rightarrow$ *software*
  - *age in [20,29] & income in [60K,100K]* $\Rightarrow$ *buying up-to-date mobile handsets*

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.

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*
- support $=$

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*
- support = P(R & data mining) = 6/100 = 0.06

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*
- support = P(R & data mining) = $6/100 = 0.06$
- confidence =

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*
- support $=$ P(R & data mining) $= 6/100 = 0.06$
- confidence $=$ support / P(R) $= 0.06/0.08 = 0.75$

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*
- support = P(R & data mining) = 6/100 = 0.06
- confidence = support / P(R) = 0.06/0.08 = 0.75
- lift =

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*
- support = P(R & data mining) = 6/100 = 0.06
- confidence = support / P(R) = 0.06/0.08 = 0.75
- lift = confidence / P(data mining) = 0.75/0.10 = 7.5

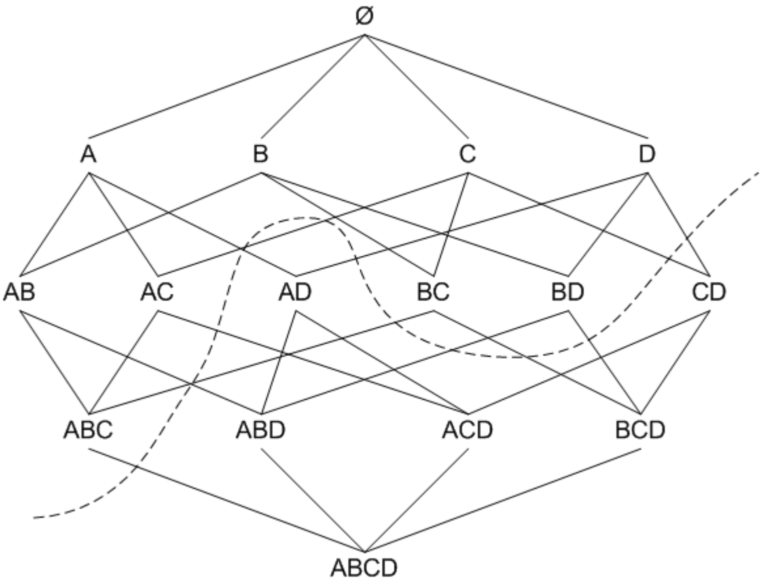
Association Rule Mining

- Association Rule Mining is normally composed of two steps:
  - Finding all frequent itemsets whose supports are no less than a minimum support threshold;
  - From above frequent itemsets, generating association rules with confidence above a minimum confidence threshold.
- The second step is straightforward, but the first one, frequent itemset generateion, is computing intensive.
- The number of possible itemsets is $2^n - 1$, where n is the number of unique items.
- Algorithms: Apriori, ECLAT, FP-Growth

Downward-Closure Property

- Downward-closure property of support, a.k.a. anti-monotonicity
- For a frequent itemset, all its subsets are also frequent.
  if {A,B} is frequent, then both {A} and {B} are frequent.
- For an infrequent itemset, all its super-sets are infrequent.
  if {A} is infrequent, then {A,B}, {A,C} and {A,B,C} are infrequent.
- useful to prune candidate itemsets

Itemset Lattice



Outline

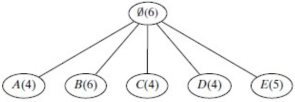
Apriori

- Apriori [Agrawal and Srikant, 1994]: a classic algorithm for association rule mining
- A level-wise, breadth-first algorithm
- Counts transactions to find frequent itemsets
- Generates candidate itemsets by exploiting downward closure property of support

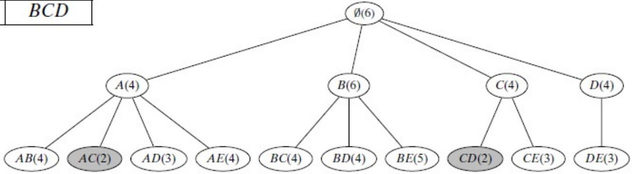
Apriori Process

1. Find all frequent 1-itemsets L_1
2. Join step: generate candidate k-itemsets by joining L_{k-1} with itself
3. Prune step: prune candidate k-itemsets using downward-closure property
4. Scan the dataset to count frequency of candidate k-itemsets and select frequent k-itemsets L_k
5. Repeat above process, until no more frequent itemsets can be found.

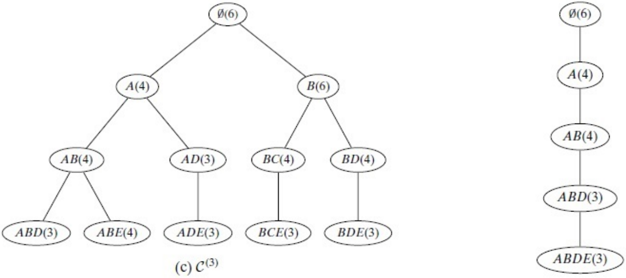
t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD



(a) $\mathcal{C}^{(1)}$



(b) $\mathcal{C}^{(2)}$



(c) $\mathcal{C}^{(3)}$

(d) $\mathcal{C}^{(4)}$

From [Zaki and Meira, 2014]

FP-growth

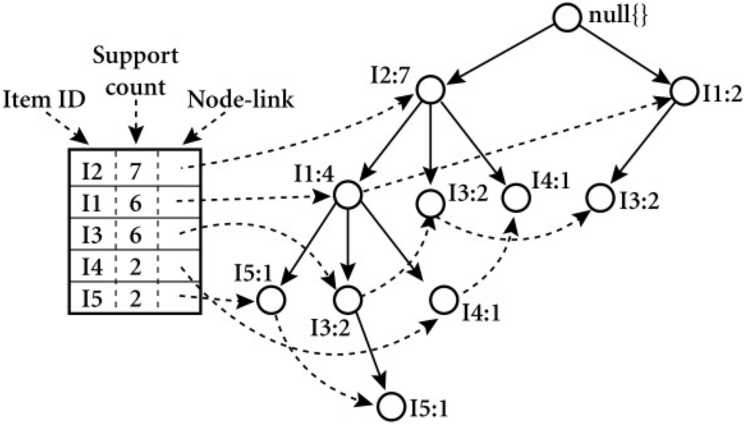
- FP-growth: frequent-pattern growth, which mines frequent itemsets without candidate generation [Han et al., 2004]
- Compresses the input database creating an FP-tree instance to represent frequent items.
- Divides the compressed database into a set of conditional databases, each one associated with one frequent pattern.
- Each such database is mined separately.
- It reduces search costs by looking for short patterns recursively and then concatenating them in long frequent patterns.†

† https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_FP-Growth_Algorithm

FP-tree

- The frequent-pattern tree (FP-tree) is a compact structure that stores quantitative information about frequent patterns in a dataset. It has two components:
  - A root labeled as "null" with a set of item-prefix subtrees as children
  - A frequent-item header table
- Each node has three attributes:
  - Item name
  - Count: number of transactions represented by the path from root to the node
  - Node link: links to the next node having the same item name
- Each entry in the frequent-item header table also has three attributes:
  - Item name
  - Head of node link: point to the first node in the FP-tree having the same item name
  - Count: frequency of the item

FP-tree



From [Han, 2005]

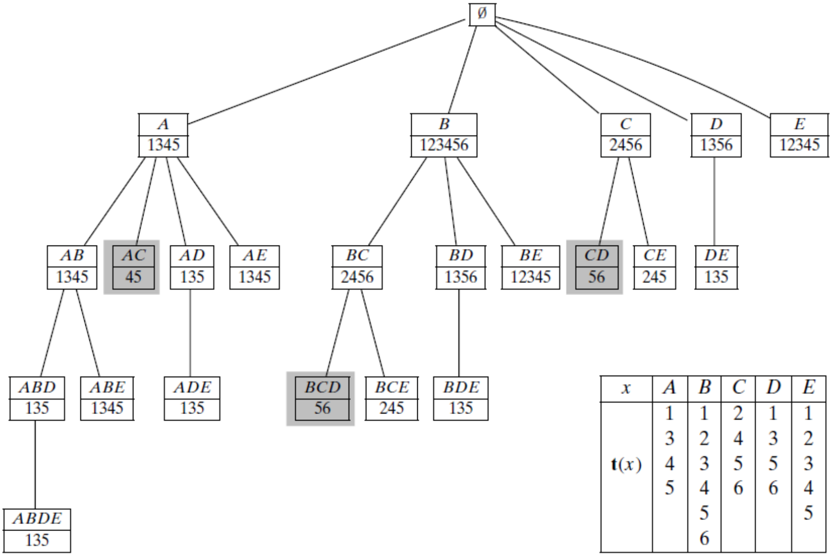
ECLAT

- ECLAT: equivalence class transformation [Zaki et al., 1997]
- A depth-first search algorithm using set intersection
- Idea: use tid set intersecion to compute the support of a candidate itemset, avoiding the generation of subsets that does not exist in the prefix tree.
- $t(AB) = t(A) \cap t(B)$
- $support(AB) = |t(AB)|$
- Eclat intersects the tidsets only if the frequent itemsets share a common prefix.
- It traverses the prefix search tree in a DFS-like manner, processing a group of itemsets that have the same prefix, also called a prefix equivalence class.

ECLAT

- It works recursively.
- The initial call uses all single items with their tid-sets.
- In each recursive call, it verifies each itemset tid-set pair $(X, t(X))$with all the other pairs to generate new candidates. If the new candidate is frequent, it is added to the set P_x.
- Recursively, it finds all frequent itemsets in the X branch.

ECLAT



x	A	B	C	D	E
$\mathbf{t}(x)$	1 3 4 5	1 2 3 4 5 6	2 4 5 6	1 3 5 6	1 2 3 4 5

From [Zaki and Meira, 2014]

Outline

Interestingness Measures

- Which rules or patterns are the most interesting ones? One way is to rank the discovered rules or patterns with interestingness measures.
- The measures of rule interestingness fall into two categories, subjective and objective [Freitas, 1998, Silberschatz and Tuzhilin, 1996].
- Objective measures, such as *lift*, *odds ratio* and *conviction*, are often data-driven and give the interestingness in terms of statistics or information theory.
- Subjective (user-driven) measures, e.g., *unexpectedness* and *actionability*, focus on finding interesting patterns by matching against a given set of user beliefs.

Objective Interestingness Measures

- Support, confidence and lift are the most widely used objective measures to select interesting rules.
- Many other objective measures introduced by Tan et al. [Tan et al., 2002], such as ϕ*-coefficient*, *odds ratio*, *kappa*, *mutual information*, *J-measure*, *Gini index*, *laplace*, *conviction*, *interest* and *cosine*.
- Their study shows that different measures have different intrinsic properties and there is no measure that is better than others in all application domains.
- In addition, any-confidence, all-confidence and bond, are designed by Omiecinski [Omiecinski, 2003].
- *Utility* is used by Chan et al. [Chan et al., 2003] to find top-k objective-directed rules.
- *Unexpected Confidence Interestingness* and *Isolated Interestingness* are designed by Dong and Li [Dong and Li, 1998] by considering its unexpectedness in terms of other association rules in its neighbourhood.

Subjective Interestingness Measures

- *Unexpectedness* and *actionability* are two main categories of subjective measures [Silberschatz and Tuzhilin, 1995].
- A pattern is unexpected if it is new to a user or contradicts the user's experience or domain knowledge.
- A pattern is actionable if the user can do something with it to his/her advantage [Silberschatz and Tuzhilin, 1995, Liu et al., 2003].
- Liu and Hsu [Liu and Hsu, 1996] proposed to rank learned rules by matching against expected patterns provided by the user.
- Ras and Wieczorkowska [Ras and Wieczorkowska, 2000] designed action-rules which show "what actions should be taken to improve the profitability of customers". The attributes are grouped into "hard attributes" which cannot be changed and "soft attributes" which are possible to change with reasonable costs. The status of customers can be moved from one to another by changing the values of soft ones.

Interestingness Measures - I

#	Measure	Formula
1	ϕ-coefficient	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-max_k P(B_k)}{2-\max_j P(A_j)-\max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j) \log \frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i) \log P(A_i), -\sum_j P(B_j) \log P(B_j))}$
8	J-Measure (J)	$\max\Big(P(A,B)\log(\frac{P(B\|A)}{P(B)}) + P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),$ $P(A,B)\log(\frac{P(A\|B)}{P(A)}) + P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\Big)$
9	Gini index (G)	$\max\Big(P(A)[P(B\|A)^2 + P(\overline{B}\|A)^2] + P(\overline{A})[P(B\|\overline{A})^2 + P(\overline{B}\|\overline{A})^2]$ $-P(B)^2 - P(\overline{B})^2,$ $P(B)[P(A\|B)^2 + P(\overline{A}\|B)^2] + P(\overline{B})[P(A\|\overline{B})^2 + P(\overline{A}\|\overline{B})^2]$ $-P(A)^2 - P(\overline{A})^2\Big)$

From [Tan et al., 2002]

Interestingness Measures - II

10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B\|A), P(A\|B))$
12	Laplace (L)	$\max\left(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2}\right)$
13	Conviction (V)	$\max\left(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})}\right)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B) - P(A)P(B)$
17	Certainty factor (F)	$\max\left(\frac{P(B\|A)-P(B)}{1-P(B)}, \frac{P(A\|B)-P(A)}{1-P(A)}\right)$
18	Added Value (AV)	$\max(P(B\|A) - P(B), P(A\|B) - P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})} \times \frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\|A) - P(B), P(A\|B) - P(A))$

From [Tan et al., 2002]

Outline

Applications - I

- Market basket analysis
  - Identifying associations between items in shopping baskets, i.e., which items are frequently purchsed together
  - Can be used by retailers to understand customer shopping habits, do selective marketing and plan shelf space
- Churn analysis and selective marketing
  - Discovering demographic characteristics and behaviours of customers who are likely/unlikely to switch to other telcos
  - Identifying customer groups who are likely to purchase a new service or product
- Credit card risk analysis
  - Finding characteristics of customers who are likely to default on credit card or mortgage
  - Can be used by banks to reduce risks when assessing new credit card or mortgage applications

Applications - II

- Stock market analysis
  - Finding relationships between individual stocks, or between stocks and economic factors
  - Can help stock traders select interesting stocks and improve trading strategies
- Medical diagnosis
  - Identifying relationships between symptoms, test results and illness
  - Can be used for assisting doctors on illness diagnosis or even on treatment

Outline

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*

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 dataframe titanic.raw
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
## 894   Crew Male Adult       No
## 1139  Crew Male Adult       No
## 827   Crew Male Adult       No
## 1727  Crew Male Adult      Yes
## 749   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

```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

```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

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

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

- Did children have a higher survival rate than adults?
- 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 have a higher survival rate than adults?
- 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 about the survival rates of other classes.

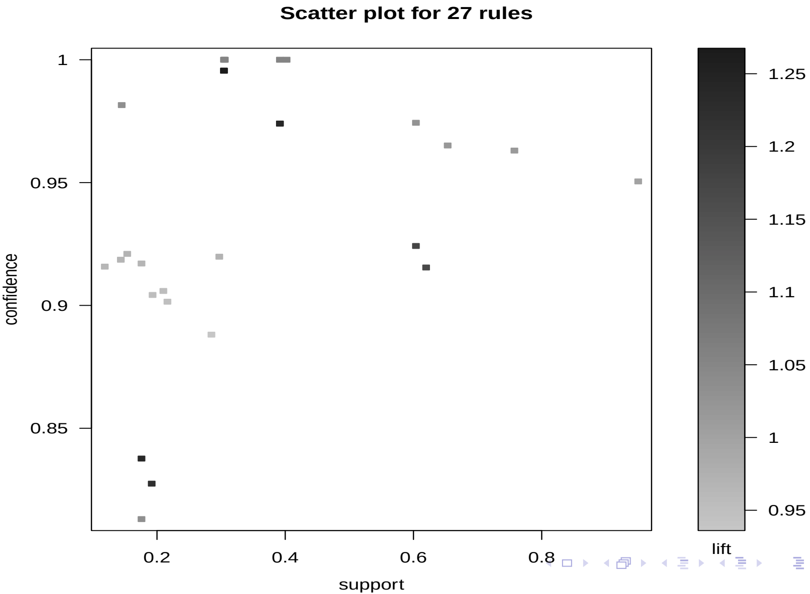
Rules about Children

```
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)
```

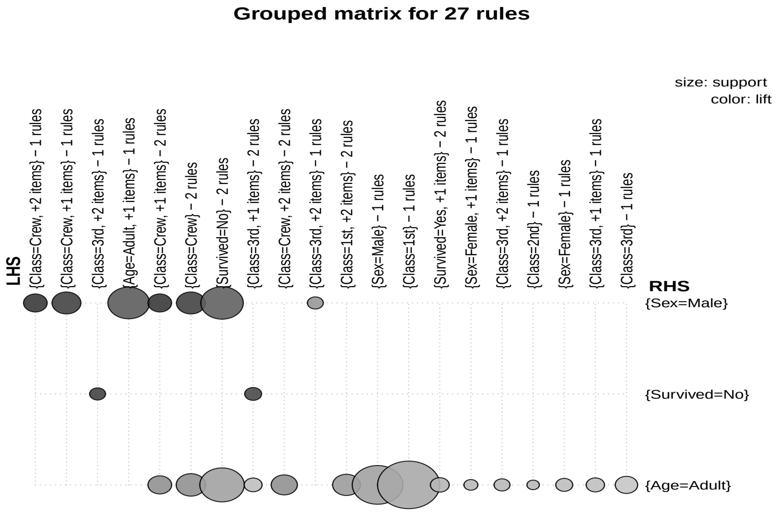
```
##   lhs              rhs                support confidence   ...
## 1 {Class=2nd,                                              ...
##    Age=Child} => {Survived=Yes} 0.010904134  1.0000000 3....
## 2 {Class=1st,                                              ...
##    Age=Child} => {Survived=Yes} 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

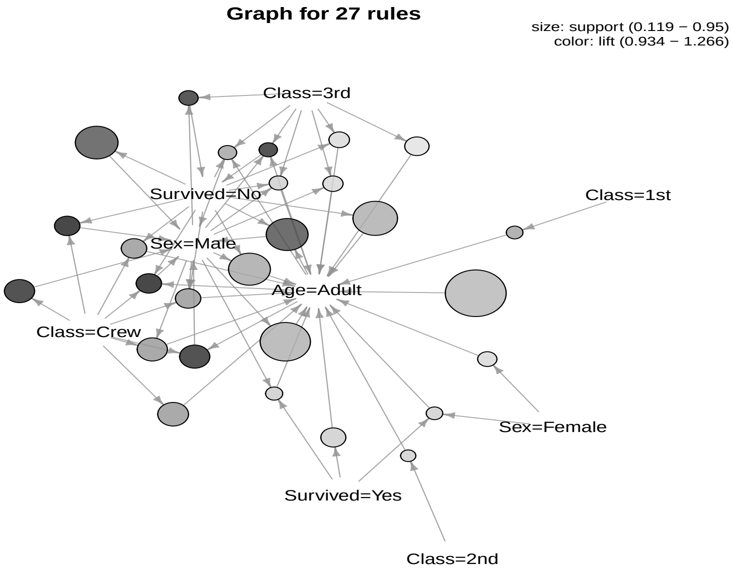
```r
library(arulesViz)
plot(rules.all)
```



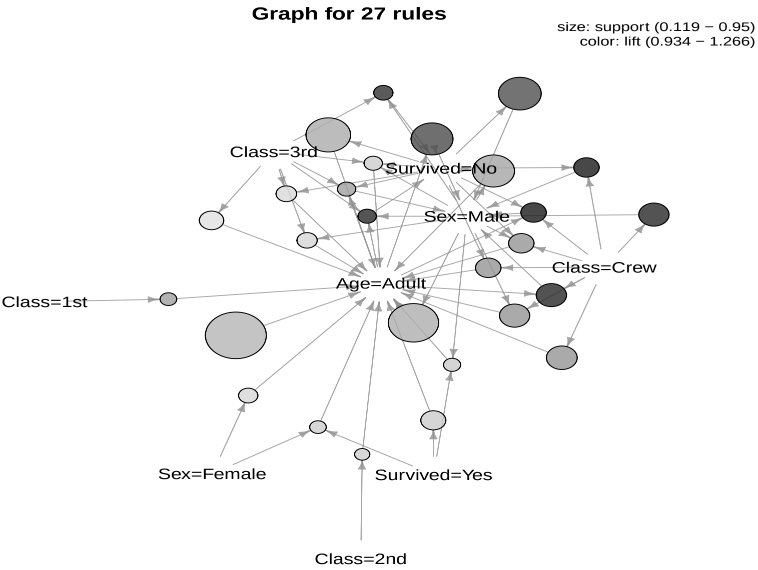
```r
plot(rules.all, method = "grouped")
```



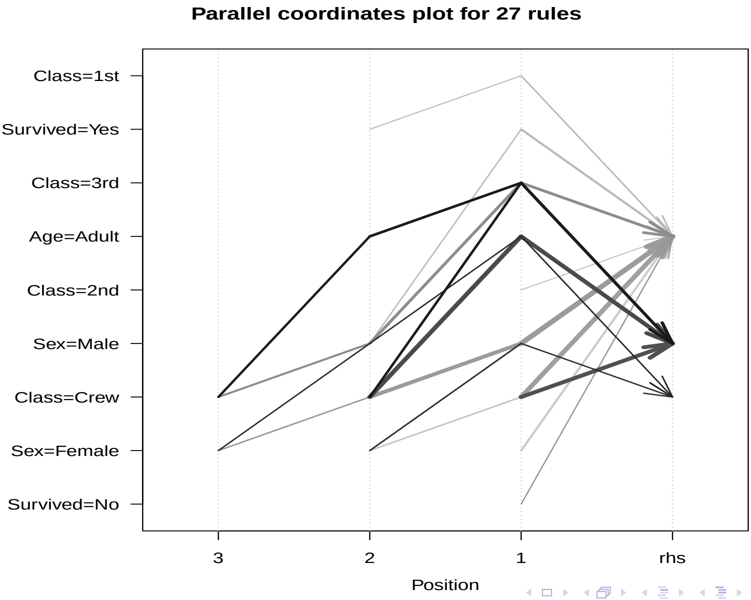
```r
plot(rules.all, method = "graph")
```



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



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



Outline

Further Readings

- Association Rule Learning
  https://en.wikipedia.org/wiki/Association_rule_learning
- 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/

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.

- More reviews on interestingness measures: [Silberschatz and Tuzhilin, 1996], [Tan et al., 2002], [Omiecinski, 2003] and [Wu et al., 2007]
- Post mining of association rules, such as selecting interesting association rules, visualization of association rules and using association rules for classification [Zhao et al., 2009]

  Yanchang Zhao, et al. (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 titled
  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 (22,000+ members)
  `http://group.rdatamining.com`
- Twitter (2,700+ followers)
  @RDataMining

References I

Agrawal, R., Imielinski, T., and Swami, A. (1993).
Mining association rules between sets of items in large databases.
In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington D.C. USA.

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.

Chan, R., Yang, Q., and Shen, Y.-D. (2003).
Mining high utility itemsets.
In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 19–26.

Dong, G. and Li, J. (1998).
Interestingness of discovered association rules in terms of neighborhood-based unexpectedness.
In *PAKDD '98: Proceedings of the Second Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining*, pages 72–86, London, UK. Springer-Verlag.

Freitas, A. A. (1998).
On objective measures of rule surprisingness.
In *PKDD '98: Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 1–9, London, UK. Springer-Verlag.

Han, J. (2005).
Data Mining: Concepts and Techniques.
Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Han, J., Pei, J., Yin, Y., and Mao, R. (2004).
Mining frequent patterns without candidate generation.
Data Mining and Knowledge Discovery, 8:53–87.

References II

Liu, B. and Hsu, W. (1996).
Post-analysis of learned rules.
In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 828–834, Portland, Oregon, USA.

Liu, B., Ma, Y., and Yu, P. S. (2003).
Discovering business intelligence information by comparing company web sites.
In Zhong, N., Liu, J., and Yao, Y. Y., editors, *Web Intelligence*, pages 105–127. Springer-Verlag.

Omiecinski, E. R. (2003).
Alternative interest measures for mining associations in databases.
IEEE Transactions on Knowledge and Data Engineering, 15(1):57–69.

Ras, Z. W. and Wieczorkowska, A. (2000).
Action-rules: How to increase profit of a company.
In *PKDD '00: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 587–592, London, UK. Springer-Verlag.

Silberschatz, A. and Tuzhilin, A. (1995).
On subjective measures of interestingness in knowledge discovery.
In *Knowledge Discovery and Data Mining*, pages 275–281.

Silberschatz, A. and Tuzhilin, A. (1996).
What makes patterns interesting in knowledge discovery systems.
IEEE Transactions on Knowledge and Data Engineering, 8(6):970–974.

Tan, P.-N., Kumar, V., and Srivastava, J. (2002).
Selecting the right interestingness measure for association patterns.
In *KDD '02: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 32–41, New York, NY, USA. ACM Press.

References III

Wu, T., Chen, Y., and Han, J. (2007).
Association mining in large databases: A re-examination of its measures.
In *PKDD'07: Proc. of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007*, pages 621–628.

Zaki, M. and Meira, W. (2014).
Data Mining and Analysis: Fundamental Concepts and Algorithms.
Cambridge University Press.

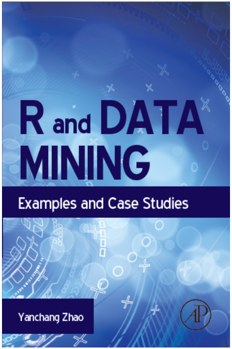
Zaki, M. J., Parthasarathy, S., Ogihara, M., and Li, W. (1997).
New algorithms for fast discovery of association rules.
Technical Report 651, Computer Science Department, University of Rochester, Rochester, NY 14627.

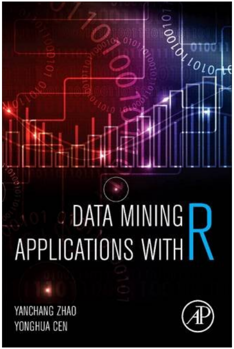
Zhao, Y. (2012).
R and Data Mining: Examples and Case Studies.
Academic Press, Elsevier.

Zhao, Y., Zhang, C., and Cao, L., editors (2009).
Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction, ISBN 978-1-60566-404-0.

Information Science Reference, Hershey, PA.

The End





Thanks!

Email: `yanchang(at)RDataMining.com`
Twitter: `@RDataMining`