
# Lecture 5: Association Rule Mining

Pilsung Kang

School of Industrial Management Engineering

Korea University

# AGENDA

# Type of Machine Learning/Data Mining

- According to the existence of target (Y) variable
  - ✓ Supervised learning vs. Unsupervised learning

### Supervised Learning

A given dataset **X** & **Y**

| | Var. 1 | Var. 2 | ... | Var. d | | Y |
|---|---|---|---|---|---|---|
| Ins. 1 | .. | .. | ... | .. | | .. |
| Ins. 2 | .. | .. | ... | .. | $y = f(x)$ | .. |
| ... | ... | ... | ... | ... | | ... |
| Ins. N | .. | .. | .. | .. | | .. |

### Unsupervised Learning

A given dataset **X**

| | Var. 1 | Var. 2 | ... | Var. d |
|---|---|---|---|---|
| Ins. 1 | .. | .. | ... | .. |
| Ins. 2 | .. | .. | ... | .. |
| ... | ... | ... | ... | ... |
| Ins. N | .. | .. | .. | .. |

### Semi-supervised Learning

A given dataset **X** & **Y**

| | Var. 1 | Var. 2 | ... | Var. d | | Y |
|---|---|---|---|---|---|---|
| Ins. 1 | .. | .. | ... | .. | | .. |
| Ins. 2 | .. | .. | ... | .. | $y = f(x)$ | .. |
| ... | ... | ... | ... | ... | | ... |
| Ins. N | .. | .. | .. | .. | | .. |
| ... | ... | ... | ... | ... | | |
| ... | ... | ... | ... | ... | | |
| ... | ... | ... | ... | ... | | |
| ... | ... | ... | ... | ... | | |
| Ins. M | .. | .. | .. | .. | | |

# Type of Machine Learning/Data Mining

- Unsupervised Learning

$$\mathcal{X} = \{\mathbf{x}_i | i = 1, \ldots, n, \ \mathbf{x}_i \in \mathbb{R}^d\}$$

**Unsupervised learning**

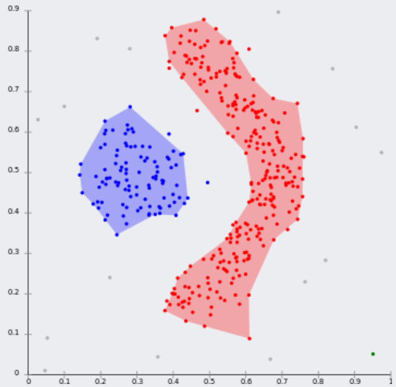
- Explores intrinsic characteristics.
- Estimates underlying distribution.
- Density estimation, clustering, novelty detection, etc.

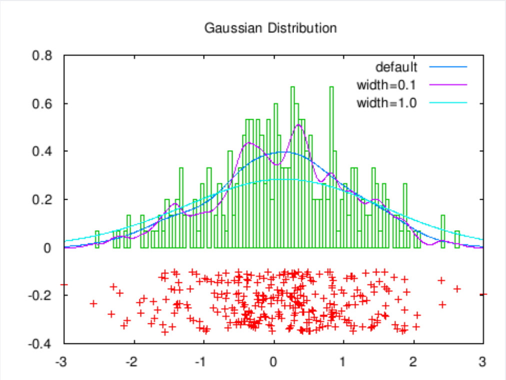
# Type of Machine Learning/Data Mining

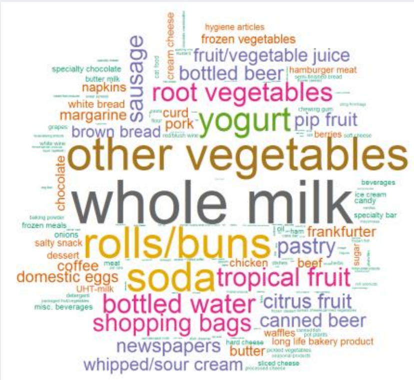
- Unsupervised Learning

A given dataset **X**

| | Var. 1 | Var. 2 | … | Var. d |
|---|---|---|---|---|
| Ins. 1 | .. | .. | … | .. |
| Ins. 2 | .. | .. | … | .. |
| … | … | … | … | … |
| Ins. N | .. | .. | .. | .. |







# Type of Machine Learning/Data Mining

- Supervised Learning

$$\mathcal{X} = \{\mathbf{x}_i | i = 1, \ldots, n, \ \mathbf{x}_i \in \mathbb{R}^d\}$$

$y = f(x)$

**Supervised learning**

- Finds relations between X and Y.
- Estimate the underlying function $y = f(x)$.
- Classification, regression.

$$\mathcal{Y} = \{y_i | i = 1, \ldots, n, \ y_i = f(\mathbf{x}_i)\}$$

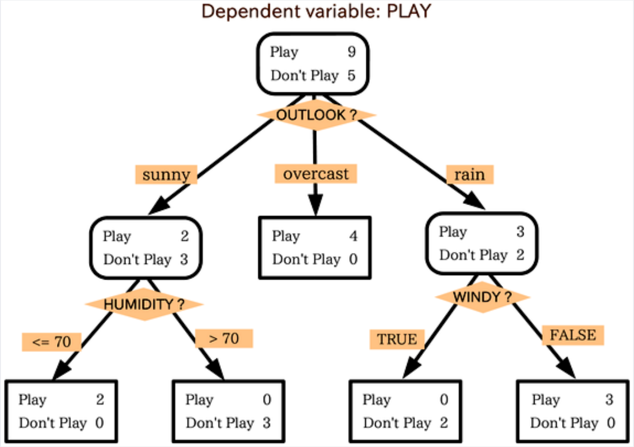
# Type of Machine Learning/Data Mining

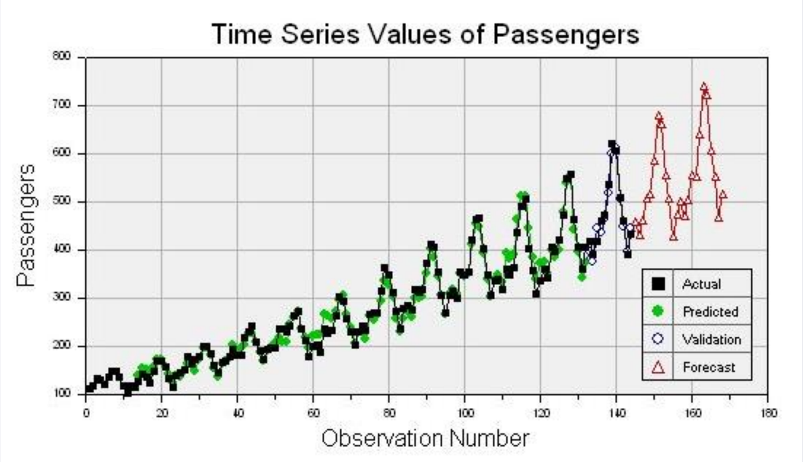
- Supervised Learning

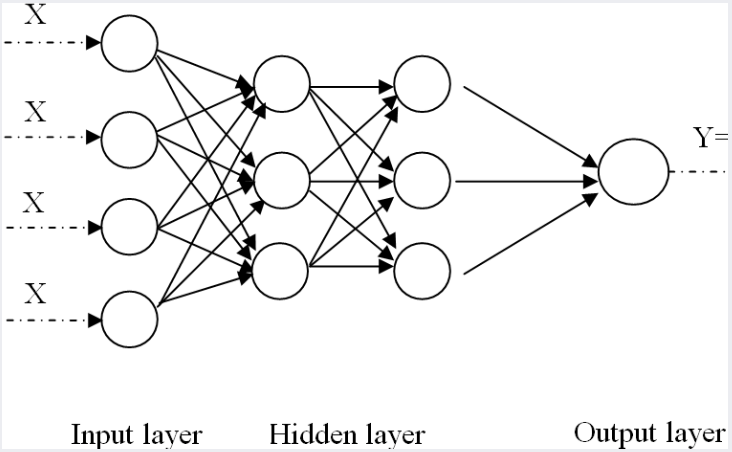
| | Var. 1 | Var. 2 | … | Var. d |
|---|---|---|---|---|
| Ins. 1 | .. | .. | … | .. |
| Ins. 2 | .. | .. | … | .. |
| … | … | … | … | … |
| Ins. N | .. | .. | .. | .. |

$y = f(x)$

| Y |
|---|
| .. |
| .. |
| … |
| .. |

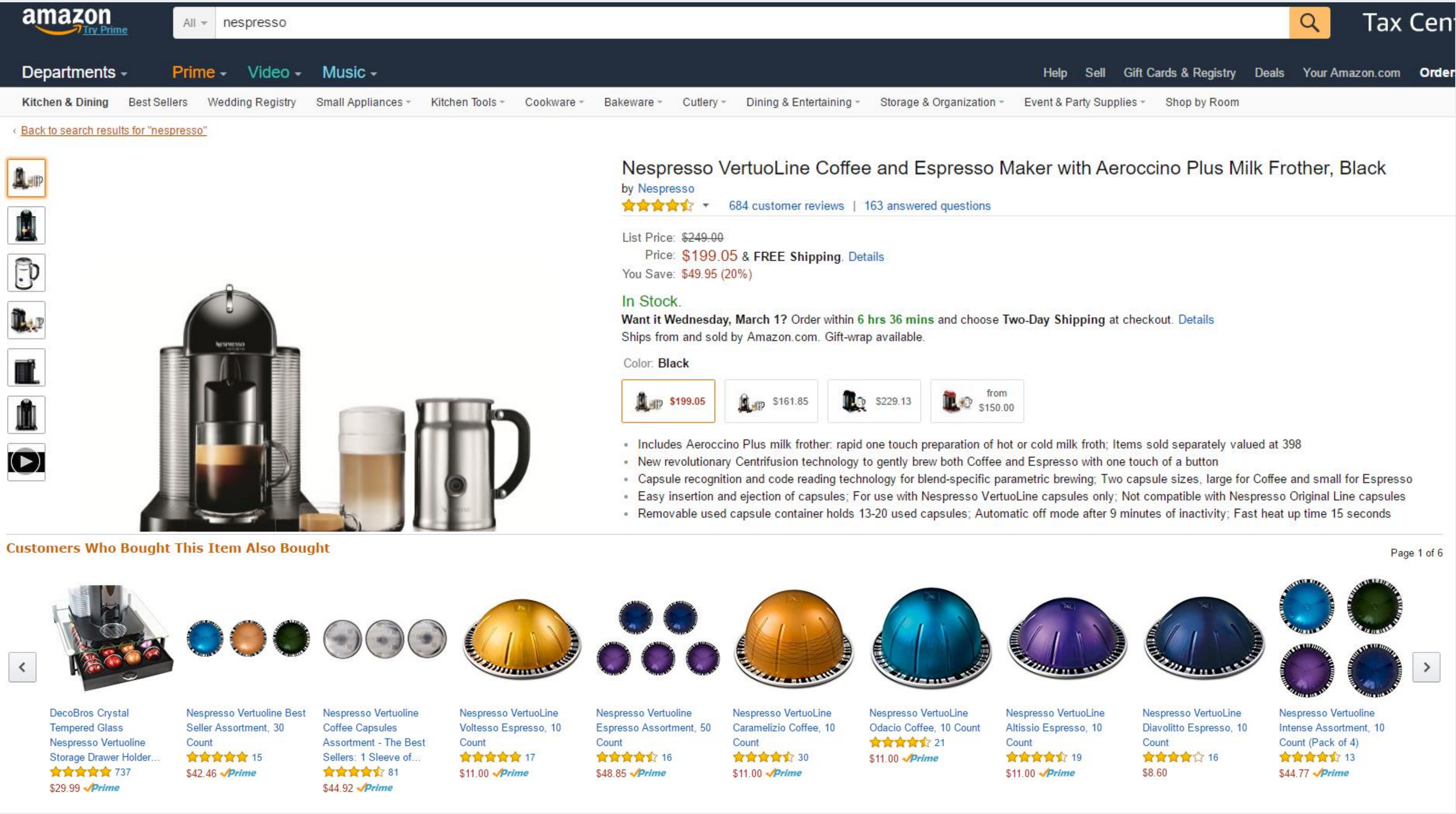


# Association Rule Mining

# Association Rule Mining



# Association Rule Mining

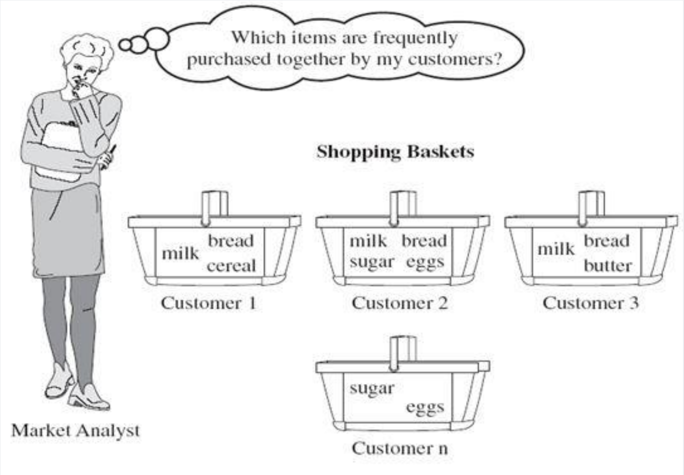
- Also known as "Market Basket Analysis"


Wall Mart (USA)


E-Mart (Korea)

# Association Rule Mining

- Goal:
  - ✓ Produce rules that define "what goes with what"
  - ✓ "If X was purchased, then Y was also purchased"
- Features
  - ✓ Rows are transactions
  - ✓ Used in recommendation systems – "Our records show that you bought X, thus you may also like Y"
  - ✓ Also called "affinity analysis" or "market basket analysis"



# Association Rule Mining

- Dataset for association rule mining
  - ✓ Each transaction is represented as a record
  - ✓ Two representations are possible: (1) item list and (2) item matrix

| TId | Items |
|---|---|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

| TId | Bread | Milk | Diapers | Beer | Eggs | Cola |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 | 1 | 1 | 0 |
| 3 | 0 | 1 | 1 | 1 | 0 | 1 |
| 4 | 1 | 1 | 1 | 1 | 0 | 0 |
| 5 | 1 | 1 | 1 | 0 | 0 | 1 |

# Association Rule Mining

- A toy example: a tiny retail market data

| Transaction | Item 1 | Item 2 | Item 3 | Item 4 |
|---|---|---|---|---|
| 1 |  |  |  | |
| 2 |  |  | | |
| 3 |  |  | | |
| 4 |  |  |  | |
| 5 |  |  | | |
| 6 |  |  | | |
| 7 |  |  | | |
| 8 |  |  |  |  |
| 9 |  |  |  | |
| 10 |  | | | |

# Association Rule Mining

- Terminology
  - ✓ Antecedent – "IF" part
  - ✓ Consequent – "THEN" part
  - ✓ Item set – the items comprising the antecedent or consequent
  - ✓ Antecedent and consequent are disjoint (have no items in common)

- Generating rules
  - ✓ Many rules are possible (e.g., for transaction 1)
    - If egg is bought, then noddle is also bought
    - If egg and noddle are bought, then tuna is also bought
    - If tuna is bought, then egg is also bought, etc.

# Association Rule Mining

Performance Measures for the rule A → B

- Support

$$\text{support}(A) = P(A)$$

  ✓ Used to find the frequent item sets

- Confidence

$$confidence(A \rightarrow B) = \frac{P(A,B)}{P(A)}$$

  ✓ Used to generate meaningful rules

- Lift

$$lift(A \rightarrow B) = \frac{P(A,B)}{P(A) \cdot P(B)}$$

  ✓ Used to determine the usefulness of generated rules

# Association Rule Mining

- How to generate an effective association rules?
  - ✓ Ideally, create all possible combinations of items and see what rules are effective and what rules are not.
  - ✓ Computation time grows exponentially as the number of items increases.

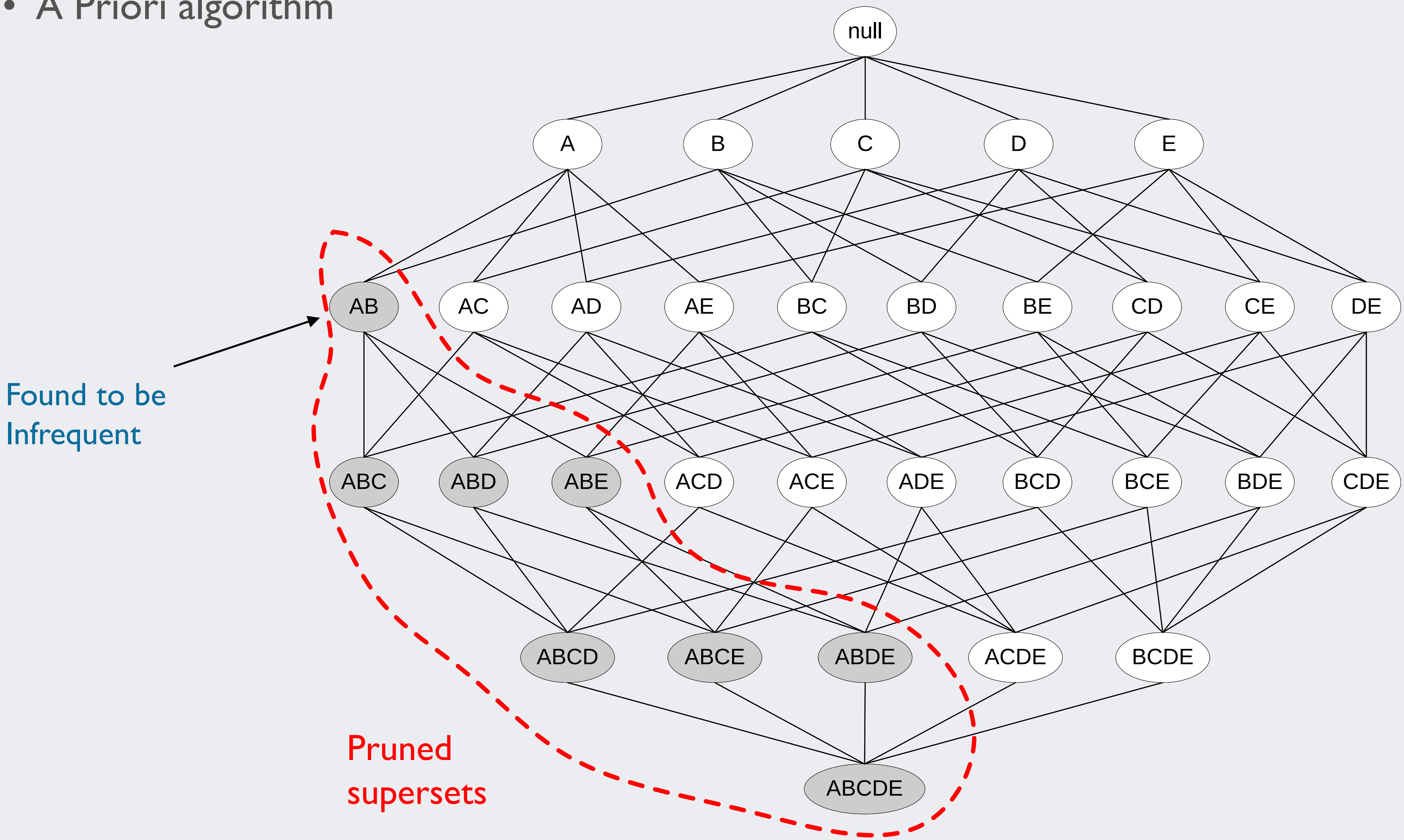
- Brute-force approach
  - ✓ List all possible association rules
  - ✓ Compute the support and confidence for each rule
  - ✓ Prune rules that fail the **minsup** and **minconf** threshold.
  - ✓ Computationally prohibitive!

# Association Rule Mining

- A priori algorithm
  - ✓ Consider only "frequent item sets"
  - ✓ "support"
    - Criterion for item set frequency P(A)
    - #(%) of transactions that include both the antecedent and the consequent
    - Support for the item set {egg, noodle} is 4 out of transactions, or 40%
  - ✓ Support of an itemset never exceeds the support of its subsets, which is known as anti-monotone property of support.

# Association Rule Mining

- A Priori algorithm



# Association Rule Mining

- Generating frequent item sets
  - ✓ Users set a minimum support criterion: e.g. 2 transactions or 20%

| Transaction | Item 1 | Item 2 | Item 3 | Item 4 |
|---|---|---|---|---|
| 1 |  |  |  | |
| 2 |  |  | | |
| 3 |  |  | | |
| 4 |  |  |  | |
| 5 |  |  | | |
| 6 |  |  | | |
| 7 |  |  | | |
| 8 |  |  |  |  |
| 9 |  |  |  | |
| 10 |  | | | |

# Association Rule Mining

- Generating frequent item sets
  - ✓ Generate the list of one-item sets that meets the support criterion
    - Support {noodle} = 8/10 = 80%
    - Support {egg} = 5/10 = 50%
    - Support {cola} = 5/10 = 50%
    - Support {rice} = 3/10 = 30%
    - Support {tuna} = 2/10 = 20%
    - Support {onion} = 1/10 = 10%
  - ✓ Onion is removed because it does not meet the minimum support criterion

# Association Rule Mining

- Generating frequent item sets
  - ✓ Use the life of one-item sets to generate list of two-item sets that meet the support criterion

| | noodle | egg | cola | rice | tuna |
|---|---|---|---|---|---|
| noodle | | 40% | 40% | 20% | 20% |
| egg | | | 30% | 0% | 20% |
| cola | | | | 0% | 10% |
| rice | | | | | 0% |
| tuna | | | | | |

  - ✓ {noodle, egg}, {noodle, cola}, {noodle, rice}, {noodle, tuna}, {egg, cola}, {egg, tuna} are frequent two-item sets

# Association Rule Mining

- Generating frequent item sets
  - ✓ Use the list of two-item sets to generate the three-item sets.
  - ✓ Continue up through k-item sets.

| Set-size | Item 1 | Item 2 | Item 3 | … | Item 6 |
|---|---|---|---|---|---|
| 1 | noodle | | | | |
| 1 | egg | | | | |
| 1 | cola | | | | |
| 1 | rice | | | | |
| 1 | tuna | | | | |
| 2 | noodle | egg | | | |
| 2 | noodle | cola | | | |
| 2 | noodle | rice | | | |
| … | … | … | | | |

# Association Rule Mining

- A Priori algorithm
  - ✓ Let k=1
  - ✓ Generate frequent itemsets of length 1
  - ✓ Repeat until no new frequent itemsets are identified
    - Generate length (k+1) candidate itemsets from length k frequent itemsets
    - Prune candidate itemsets containing subsets of length k that are infrequent
    - Count the support of each candidate by scanning the DB
    - Eliminate candidates that are infrequent, leaving only those that are frequent

# Association Rule Mining

- Confidence
  - ✓ The % of antecedent transactions that also have the consequent item set
  - ✓ E.g. "if noddle is purchased, then egg is also purchased"

$$\text{support}(noodle) = P(noodle) = \frac{8}{10}, \quad \text{support}(egg) = P(egg) = \frac{5}{10}$$

$$confidence(noodle \rightarrow egg) = \frac{P(noodle, egg)}{P(noodle)} = \frac{4/10}{8/10} = 0.5(50\%)$$

  - ✓ Benchmark confidence: transactions with consequent (P(egg), support(egg)) of all transactions
  - ✓ If the confidence of (noodle → egg) is smaller then the support of egg, this rule is useless

# Association Rule Mining

- Lift
  - ✓ Confidence/(benchmark confidence)

$$lift(noodle \to egg)$$

$$= \frac{confidence(noodle \to egg)}{\text{support}(egg)} = \frac{\frac{P(noodle, egg)}{P(noodle)}}{P(egg)} = \frac{P(noodle, egg)}{P(noodle) \times P(egg)}$$

$$= \frac{\frac{4}{10}}{\frac{8}{10} \times \frac{5}{10}} = 1$$

- ✓ If lift = 1, then the antecedent and the consequents are statistically independent
- ✓ If lift > 1, then the rule is useful in finding consequent item sets

# Association Rule Mining

- Generated rules
  - ✓ Set the support to 20%.
  - ✓ Set the confidence to 70%.

| Rule # | Antecedent (a) | Consequent | Support | Confidence | Lift |
|---|---|---|---|---|---|
| 1 | tuna=> | egg, noodle | 2 | 100 | 2.5 |
| 2 | tuna=> | egg | 2 | 100 | 2 |
| 3 | noodle, tuna=> | egg | 2 | 100 | 2 |
| 4 | rice=> | noodle | 3 | 100 | 1.25 |
| 5 | egg, tuna=> | noodle | 2 | 100 | 1.25 |
| 6 | tuna=> | noodle | 2 | 100 | 1.25 |
| 7 | cola=> | noodle | 5 | 80 | 1 |
| 8 | egg=> | noodle | 5 | 80 | 1 |

# Association Rule Mining

- Summary
  - ✓ Produce rules on associations between items from a database of transactions
  - ✓ Widely used in recommender systems
  - ✓ Most popular method is A-priori algorithm
  - ✓ To reduce computation, consider only "frequent" item sets (=support)
  - ✓ Performance is measured by confidence and lift

# AGENDA

# Association Rule Mining

- Package "arules" & "arulesViz

## Package 'arules'

October 2, 2016

**Version** 1.5-0

**Date** 2016-09-23

**Title** Mining Association Rules and Frequent Itemsets

**Description** Provides the infrastructure for representing, manipulating and analyzing transaction data and patterns (frequent itemsets and association rules). Also provides interfaces to C implementations of the association mining algorithms Apriori and Eclat by C. Borgelt.

**Classification/ACM** G.4, H.2.8, I.5.1

**URL** http://lyle.smu.edu/IDA/arules

**BugReports** https://github.com/mhahsler/arules/issues

## Package 'arulesViz'

October 3, 2016

**Version** 1.2-0

**Date** 2016-10-02

**Title** Visualizing Association Rules and Frequent Itemsets

**Author** Michael Hahsler and Sudheer Chelluboina

**Maintainer** Michael Hahsler <mhahsler@lyle.smu.edu>

**Depends** arules (>= 1.4.1), grid

**Imports** scatterplot3d, vcd, seriation, igraph (>= 1.0.0), graphics, methods, utils, grDevices, stats, colorspace, DT, plotly

**Suggests** graph, Rgraphviz, iplots

**Description**
Extends package arules with various visualization techniques for association rules and itemsets. The package also includes several interactive visualizations for rule exploration.

**License** GPL-3

**URL** http://lyle.smu.edu/IDA/arules/

**BugReports** https://github.com/mhahsler/arulesViz/issues

# Association Rule Mining

- Titanic Data

```r
# Association Rules ------------------------------------------------------
# arules and arulesViz packages install
install.packages("arules", dependencies = TRUE)
install.packages("arulesViz", dependencies = TRUE)

library(arules)
library(arulesViz)
library(wordcloud)

# Load titanic data set
titanic <- read.delim("titanic.txt", dec=",")
str(titanic)
head(titanic)
```

```
> # Load titanic data set
> titanic <- read.delim("titanic.txt", dec=",")
> str(titanic)
'data.frame':	1313 obs. of  5 variables:
 $ Name    : Factor w/ 1310 levels "Abbing, Mr Anthony",..: 22 25 26 27 24 31 45 46 50 54 ...
 $ PClass  : Factor w/ 3 levels "1st","2nd","3rd": 1 1 1 1 1 1 1 1 1 1 ...
 $ Age     : Factor w/ 75 levels "0.17","0.33",..: 28 18 30 24 5 48 66 39 60 73 ...
 $ Sex     : Factor w/ 2 levels "female","male": 1 1 2 1 2 2 1 2 1 2 ...
 $ Survived: int  1 0 0 0 1 1 1 0 1 0 ...
> head(titanic)
                                          Name PClass  Age    Sex Survived
1                 Allen, Miss Elisabeth Walton    1st   29 female        1
2                  Allison, Miss Helen Loraine    1st    2 female        0
3          Allison, Mr Hudson Joshua Creighton    1st   30   male        0
4 Allison, Mrs Hudson JC (Bessie Waldo Daniels)    1st   25 female        0
5                Allison, Master Hudson Trevor    1st 0.92   male        1
6                           Anderson, Mr Harry    1st   47   male        1
```

# Association Rule Mining

- Data Preprocessing
  - ✓ Categorize a numeric variable, remove NA, etc.

```r
# Remove "Name" column and group "Age" column
titanic_ar <- titanic[,2:5]
titanic_ar$Age = as.character(titanic_ar$Age)
c_idx <- which(as.numeric(titanic_ar$Age) < 20)
a_idx <- which(as.numeric(titanic_ar$Age) >= 20)
na_idx <- which(is.na(titanic_ar$Age))

titanic_ar$Age[c_idx] <- "Child"
titanic_ar$Age[a_idx] <- "Adult"
titanic_ar$Age[na_idx] <- "Unknown"

# Convert the attribues to factor
titanic_ar$Age <- as.factor(titanic_ar$Age)
titanic_ar$Survived <- as.factor(titanic_ar$Survived)
```

| | PClass | Age | Sex | Survived |
|---|---|---|---|---|
| 1 | 1st | Adult | female | 1 |
| 2 | 1st | Child | female | 0 |
| 3 | 1st | Adult | male | 0 |
| 4 | 1st | Adult | female | 0 |
| 5 | 1st | Child | male | 1 |
| 6 | 1st | Adult | male | 1 |
| 7 | 1st | Adult | female | 1 |
| 8 | 1st | Adult | male | 0 |
| 9 | 1st | Adult | female | 1 |
| 10 | 1st | Adult | male | 0 |

# Association Rule Mining

- Find rules (default setting)

```
# Rule generation by Apriori algorithm with default settings
rules <- apriori(titanic_ar)
inspect(rules)
```

```
> rules <- apriori(titanic_ar)

parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen target   ext
        0.8    0.1    1 none FALSE            TRUE     0.1      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 Borgelt
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[10 item(s), 1313 transaction(s)] done [0.00s].
sorting and recoding items ... [10 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [16 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

# Association Rule Mining

- Find rules (default setting)

```
> inspect(rules)
   lhs               rhs            support   confidence    lift
1  {PClass=3rd}   => {Survived=0} 0.4364052  0.8059072 1.226137
2  {Sex=male}     => {Survived=0} 0.5399848  0.8331375 1.267566
3  {Survived=0}   => {Sex=male}   0.5399848  0.8215527 1.267566
4  {PClass=2nd,
    Sex=male}     => {Survived=0} 0.1127190  0.8554913 1.301576
5  {PClass=2nd,
    Survived=0}   => {Sex=male}   0.1127190  0.9192547 1.418309
6  {PClass=1st,
    Sex=female}   => {Survived=1} 0.1020564  0.9370629 2.734141
7  {Sex=female,
    Survived=0}   => {PClass=3rd} 0.1005331  0.8571429 1.582881
8  {PClass=3rd,
    Age=Unknown}  => {Survived=0} 0.2536177  0.8473282 1.289156
9  {Age=Unknown,
    Sex=male}     => {Survived=0} 0.2566641  0.8798956 1.338706
10 {Age=Unknown,
    Survived=0}   => {Sex=male}   0.2566641  0.8023810 1.237986
11 {Age=Adult,
    Sex=male}     => {Survived=0} 0.2482864  0.8253165 1.255667
12 {Age=Adult,
    Survived=0}   => {Sex=male}   0.2482864  0.8693333 1.341286
13 {PClass=3rd,
    Sex=male}     => {Survived=0} 0.3358720  0.8837675 1.344596
14 {PClass=3rd,
    Age=Unknown,
    Sex=male}     => {Survived=0} 0.1957350  0.9081272 1.381658
15 {PClass=3rd,
    Age=Adult,
    Sex=male}     => {Survived=0} 0.1142422  0.8670520 1.319165
16 {PClass=3rd,
    Age=Adult,
    Survived=0}   => {Sex=male}   0.1142422  0.8064516 1.244267
```

# Association Rule Mining

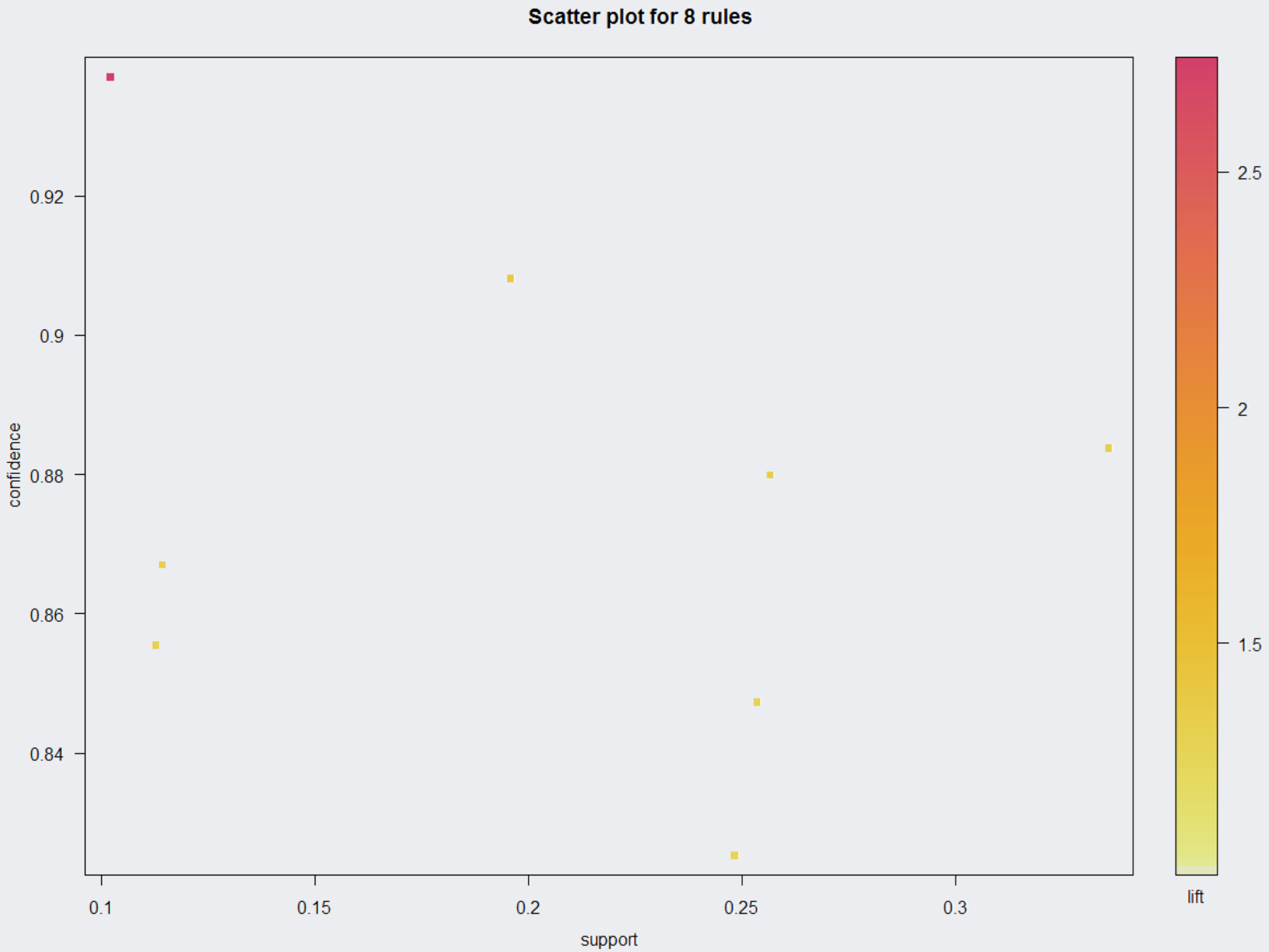
- Find rules (customized setting)

```r
# Rule generation by Apriori algorithm with custom settings
rules <- apriori(titanic_ar, parameter = list(minlen = 3, support = 0.1, conf = 0.8),
                 appearance = list(rhs = c("Survived=0", "Survived=1"), default="lhs"))
inspect(rules)

# Plot the rules
plot(rules, method="scatterplot")
plot(rules, method="graph", control=list(type = "items", alpha = 1))
plot(rules, method="paracoord", control=list(reorder=TRUE))
```

```
> inspect(rules)
  lhs                rhs            support   confidence    lift
1 {PClass=2nd,
   Sex=male}      => {Survived=0} 0.1127190  0.8554913 1.301576
2 {PClass=1st,
   Sex=female}    => {Survived=1} 0.1020564  0.9370629 2.734141
3 {PClass=3rd,
   Age=Unknown}   => {Survived=0} 0.2536177  0.8473282 1.289156
4 {Age=Unknown,
   Sex=male}      => {Survived=0} 0.2566641  0.8798956 1.338706
5 {Age=Adult,
   Sex=male}      => {Survived=0} 0.2482864  0.8253165 1.255667
6 {PClass=3rd,
   Sex=male}      => {Survived=0} 0.3358720  0.8837675 1.344596
7 {PClass=3rd,
   Age=Unknown,
   Sex=male}      => {Survived=0} 0.1957350  0.9081272 1.381658
8 {PClass=3rd,
   Age=Adult,
   Sex=male}      => {Survived=0} 0.1142422  0.8670520 1.319165
```

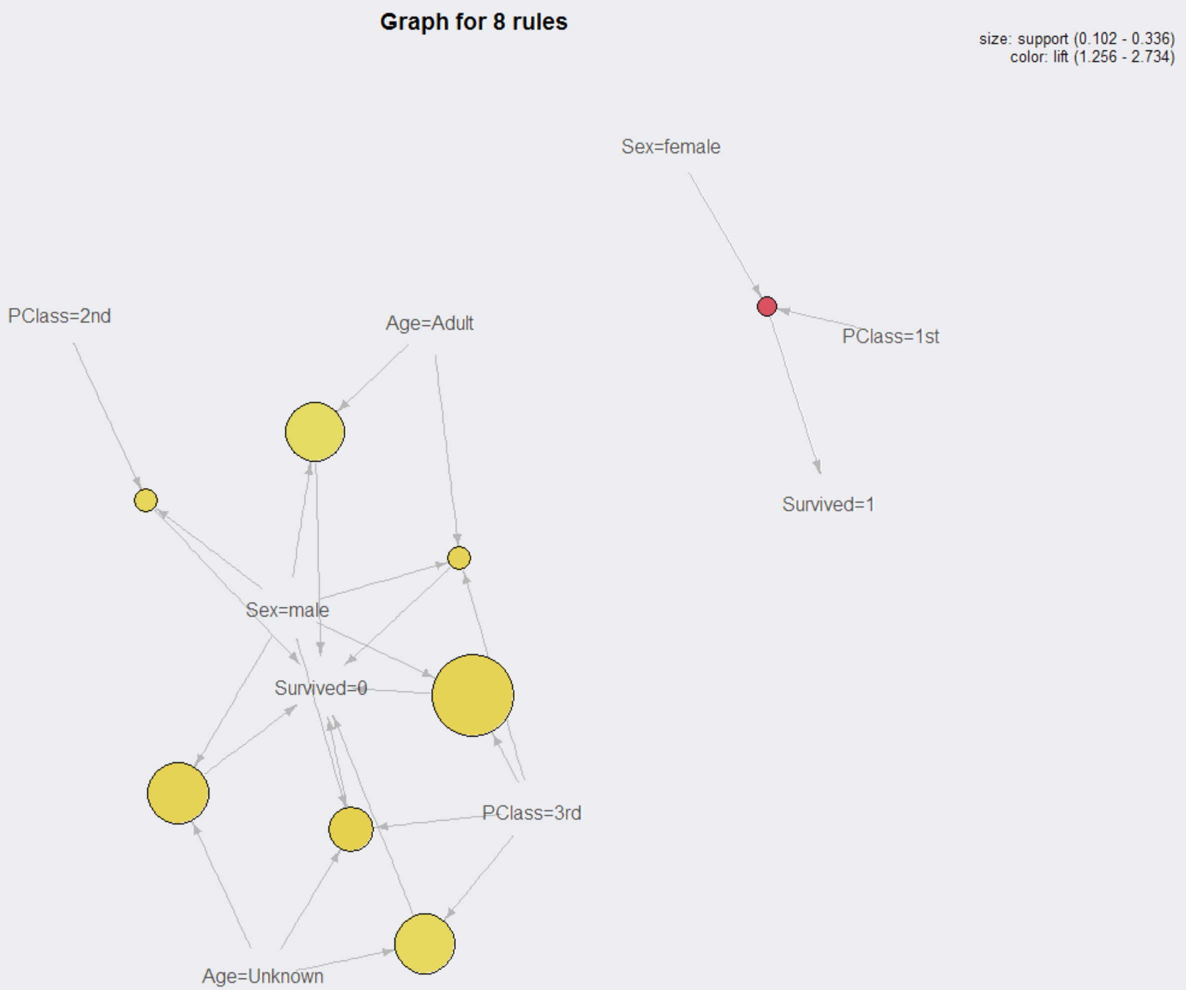
# Association Rule Mining

- Visualize the rules



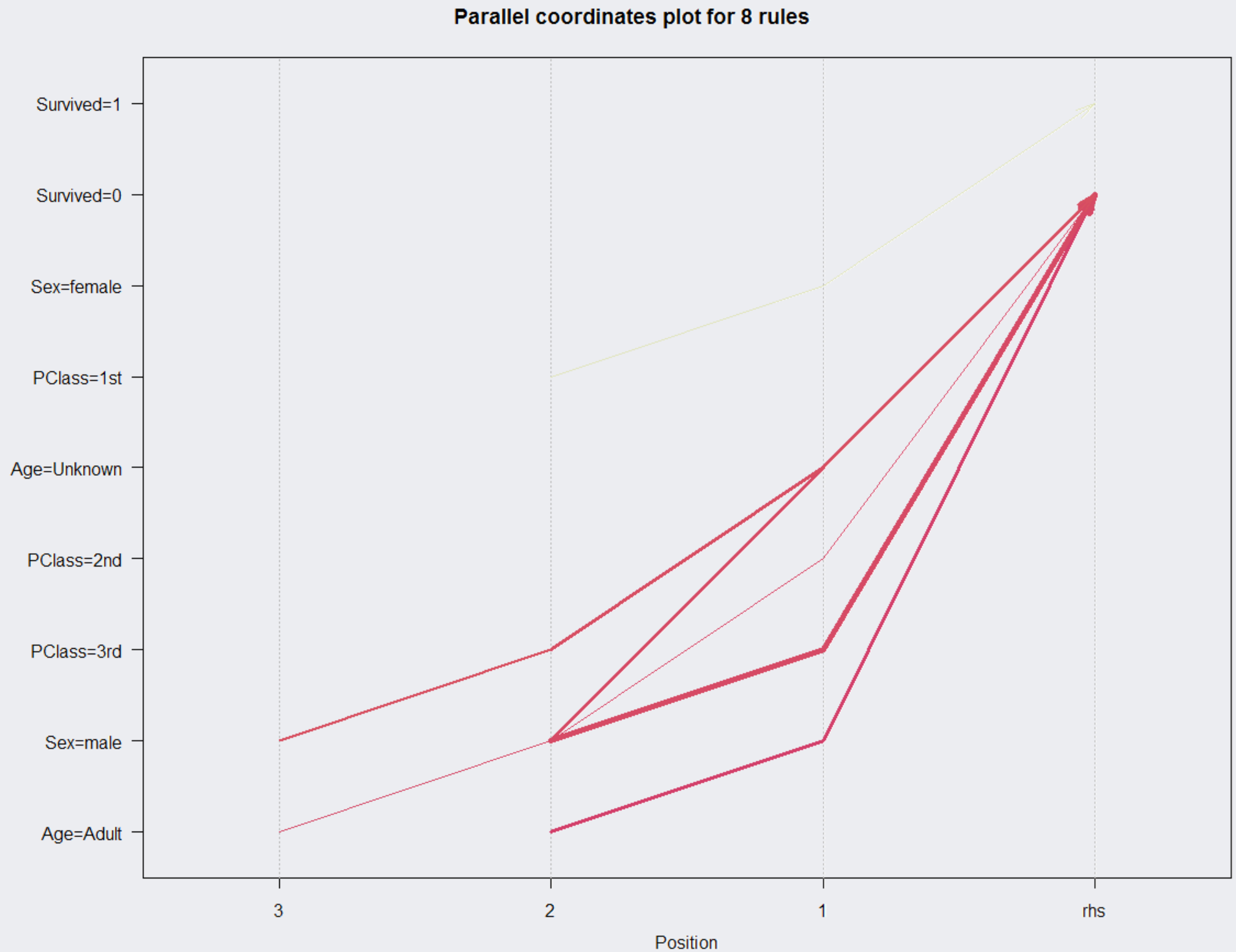
# Association Rule Mining

- Visualize the rules



# Association Rule Mining

- Visualize the rules



# Association Rule Mining

- Groceries shopping data

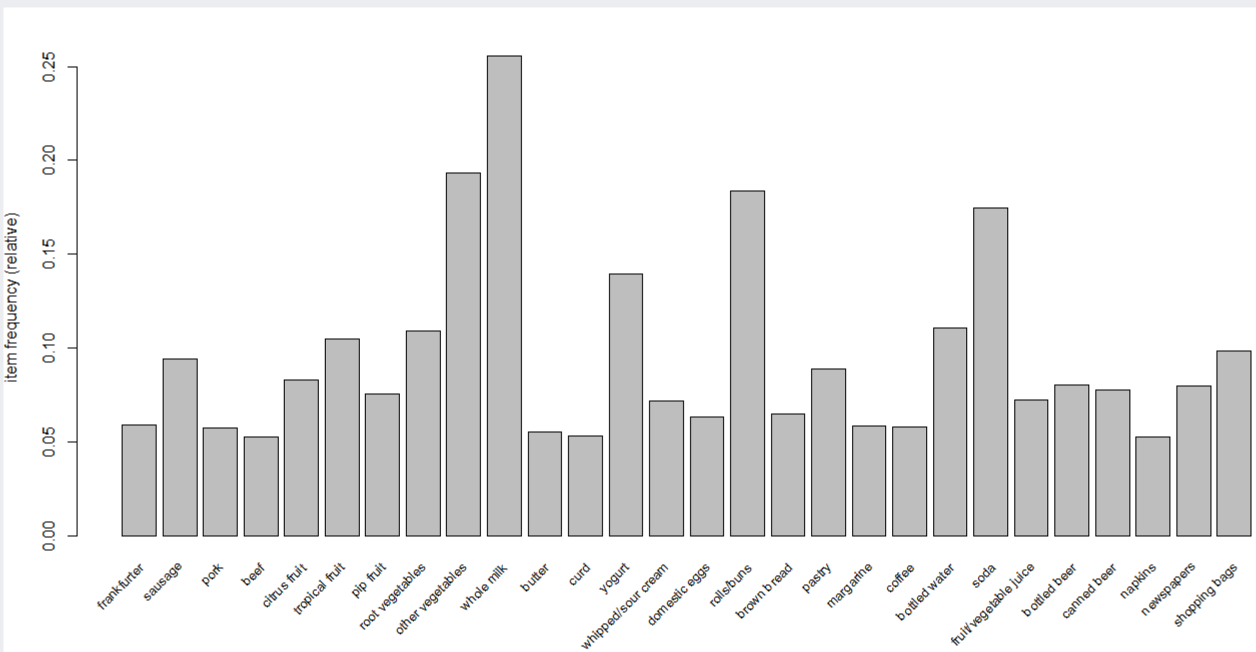
```r
# Load transaction data "Groceries"
data("Groceries")
summary(Groceries)
str(Groceries)
inspect(Groceries)
```

```
> summary(Groceries)
transactions as itemMatrix in sparse format with
 9835 rows (elements/itemsets/transactions) and
 169 columns (items) and a density of 0.02609146

most frequent items:
      whole milk other vegetables       rolls/buns             soda           yogurt          (Other)
            2513             1903             1809             1715             1372            34055

element (itemset/transaction) length distribution:
sizes
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24
2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77   55   46   29   14   14    9   11    4    6    1
  26   27   28   29   32
   1    1    1    3    1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   4.409   6.000  32.000

includes extended item information - examples:
       labels  level2           level1
1 frankfurter sausage meet and sausage
2     sausage sausage meet and sausage
3  liver loaf sausage meet and sausage
```

# Association Rule Mining

- Groceries shopping data
  - ✓ Item inspection

```
# Item inspection
itemName <- itemLabels(Groceries)
itemCount <- itemFrequency(Groceries)*9835

col <- brewer.pal(8, "Dark2")
wordcloud(words = itemName, freq = itemCount, min.freq = 1, scale = c(10, 0.2), col = col , random.order = FALSE)

itemFrequencyPlot(Groceries, support = 0.05, cex.names=0.8)
```



# Association Rule Mining

- Groceries shopping data
  - ✓ Item inspection

# Association Rule Mining

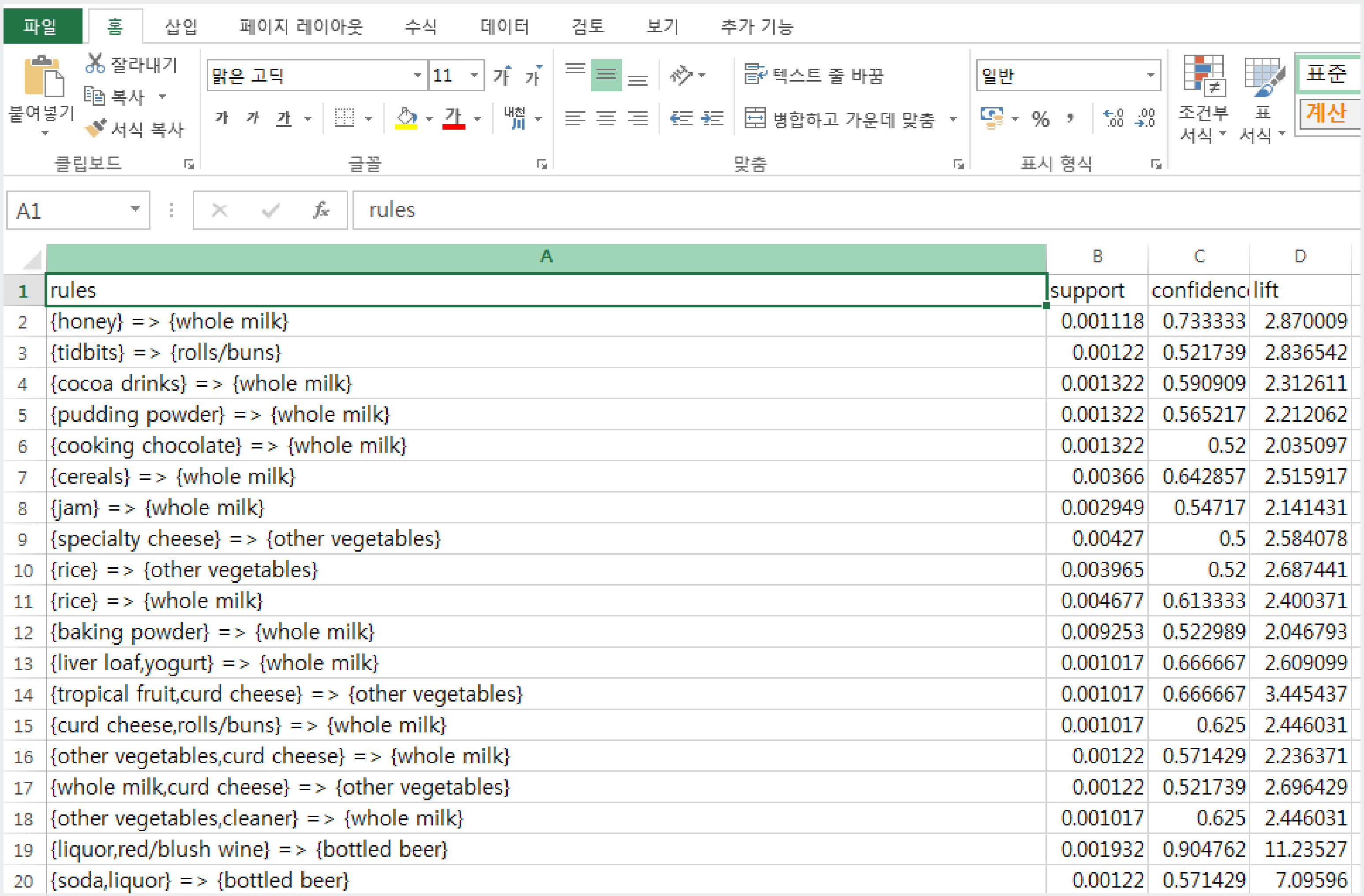
- Find and visualize rules

```
# Rule generation by Apriori
rules <- apriori(Groceries, parameter=list(support=0.001, confidence=0.5))
rules

# List the first three rules with the highest lift values
inspect(head(sort(rules, by="lift"),3))

# Save the rules in a text file
write.csv(as(rules, "data.frame"), "Groceries_rules.csv", row.names = FALSE)

# Plot the rules
plot(rules)
plot(rules, method="grouped")
```

```
> inspect(head(sort(rules, by="lift"),3))
  lhs                          rhs                  support confidence     lift
1 {Instant food products,
   soda}                    => {hamburger meat} 0.001220132  0.6315789 18.99565
2 {soda,
   popcorn}                 => {salty snack}    0.001220132  0.6315789 16.69779
3 {flour,
   baking powder}           => {sugar}          0.001016777  0.5555556 16.40807
```

# Association Rule Mining

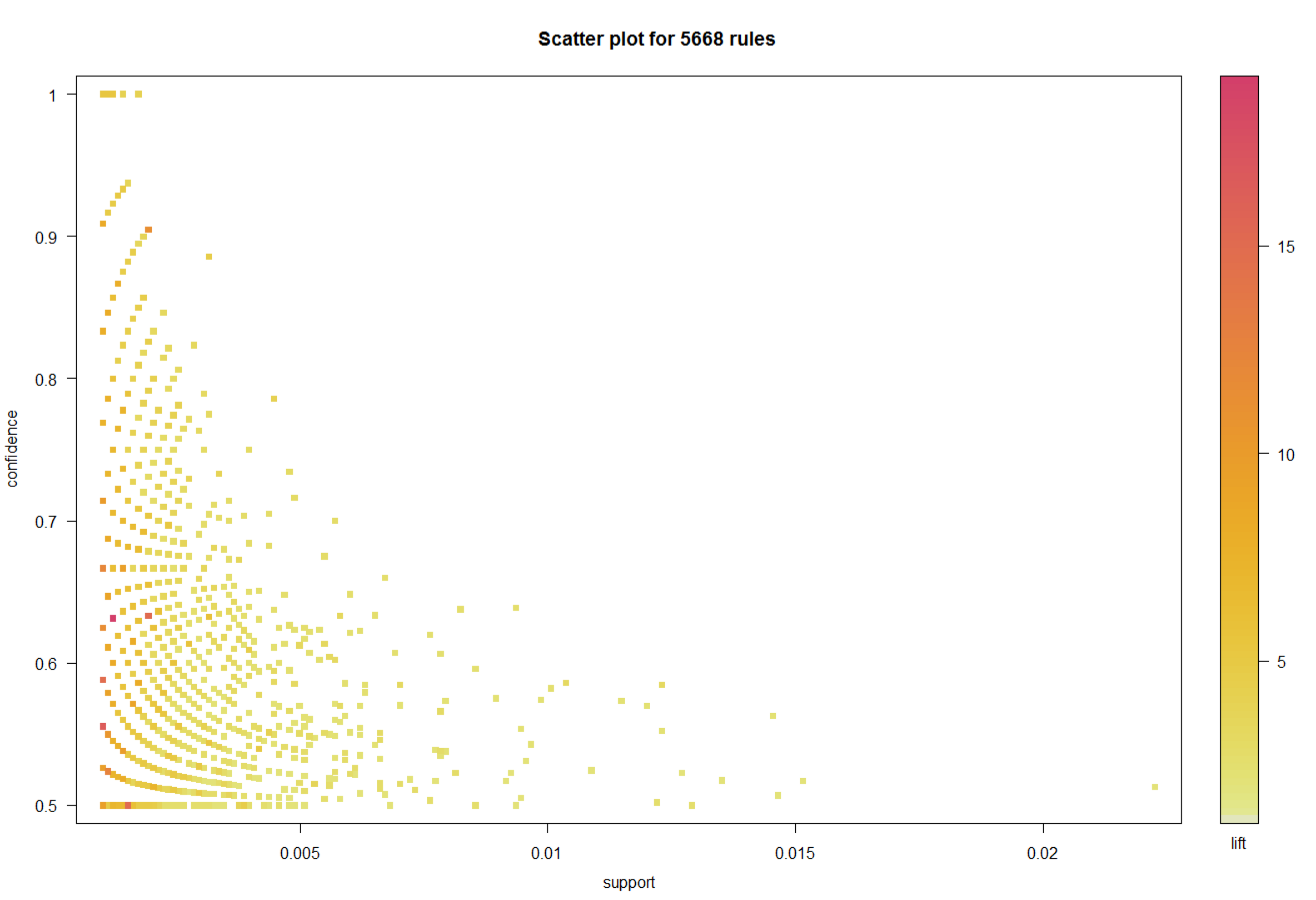
- Find and visualize rules




| | A | B | C | D |
|---|---|---|---|---|
| 1 | rules | support | confidenc | lift |
| 2 | {honey} => {whole milk} | 0.001118 | 0.733333 | 2.870009 |
| 3 | {tidbits} => {rolls/buns} | 0.00122 | 0.521739 | 2.836542 |
| 4 | {cocoa drinks} => {whole milk} | 0.001322 | 0.590909 | 2.312611 |
| 5 | {pudding powder} => {whole milk} | 0.001322 | 0.565217 | 2.212062 |
| 6 | {cooking chocolate} => {whole milk} | 0.001322 | 0.52 | 2.035097 |
| 7 | {cereals} => {whole milk} | 0.00366 | 0.642857 | 2.515917 |
| 8 | {jam} => {whole milk} | 0.002949 | 0.54717 | 2.141431 |
| 9 | {specialty cheese} => {other vegetables} | 0.00427 | 0.5 | 2.584078 |
| 10 | {rice} => {other vegetables} | 0.003965 | 0.52 | 2.687441 |
| 11 | {rice} => {whole milk} | 0.004677 | 0.613333 | 2.400371 |
| 12 | {baking powder} => {whole milk} | 0.009253 | 0.522989 | 2.046793 |
| 13 | {liver loaf,yogurt} => {whole milk} | 0.001017 | 0.666667 | 2.609099 |
| 14 | {tropical fruit,curd cheese} => {other vegetables} | 0.001017 | 0.666667 | 3.445437 |
| 15 | {curd cheese,rolls/buns} => {whole milk} | 0.001017 | 0.625 | 2.446031 |
| 16 | {other vegetables,curd cheese} => {whole milk} | 0.00122 | 0.571429 | 2.236371 |
| 17 | {whole milk,curd cheese} => {other vegetables} | 0.00122 | 0.521739 | 2.696429 |
| 18 | {other vegetables,cleaner} => {whole milk} | 0.001017 | 0.625 | 2.446031 |
| 19 | {liquor,red/blush wine} => {bottled beer} | 0.001932 | 0.904762 | 11.23527 |
| 20 | {soda,liquor} => {bottled beer} | 0.00122 | 0.571429 | 7.09596 |

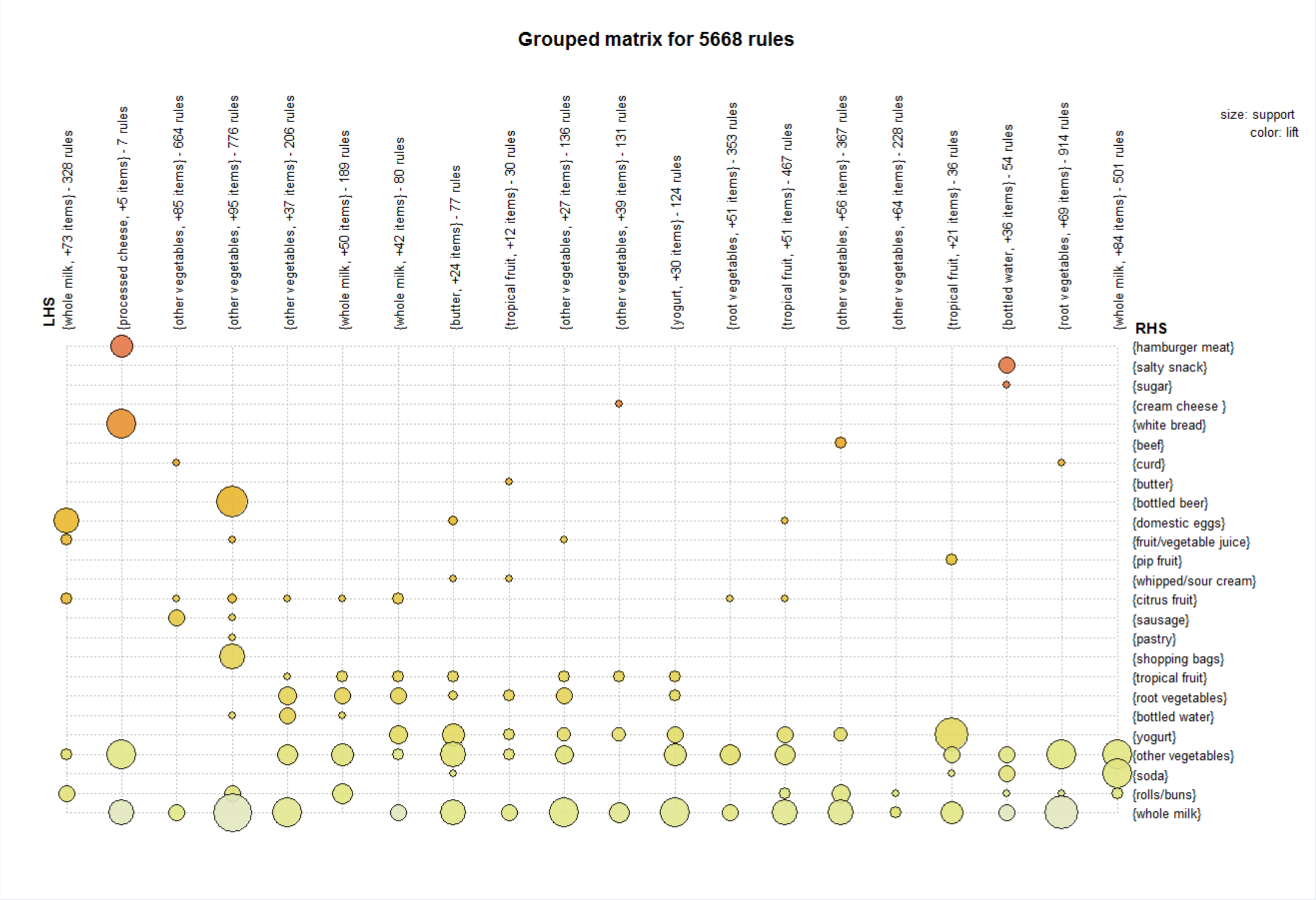
# Association Rule Mining

- Find and visualize rules



# Association Rule Mining

- Find and visualize rules




ANY
questions?