Practical: R and data mining

You may want to explore the website freely this week. Please click http://www.rdatamining.com/ link to open the resource. Below are R scripts/packages for data mining which are selected from this link.

1. Decision Tree

Building a decision tree and visualise it

```
library("party")
str(iris)

# Call function ctree to build a decision tree.
# The first parameter is a formula, which defines a target
# variable and a list of independent variables.

iris_ctree <-
  ctree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
                    , data=iris)

print(iris_ctree)
plot(iris_ctree)
plot(iris_ctree, type="simple")
```

2. Hierarchical clustering

Draw a sample of 40 records from iris data, and remove variable Species

```
idx <- sample(1:dim(iris)[1], 40)
irisSample <- iris[idx,]
irisSample$Species <- NULL
```

Perform hierarchical clustering

```
hc <- hclust(dist(irisSample), method="ave")
plot(hc, hang = -1, labels=iris$Species[idx])
```

3. Outlier detection

The below script uses the LOF (Local Outlier Factor) algorithm to detect outliers. The LOF algorithm identifies local outliers based on density. The detail of the algorithm can be seen in https://www.researchgate.net/publication/221214719_LOF_Identifying_Density-Based_Local_Outliers

```
library(DMwR2)

 # remove "Species", which is a categorical column
 iris2 <- iris[,1:4]
 outlier.scores <- lofactor(iris2, k=5)
 plot(density(outlier.scores))

 # pick top 5 as outliers
 outliers <- order(outlier.scores, decreasing=T)[1:5]
 # who are outliers
 print(outliers)
 print(iris2[outliers,])
```

show outliers with a pairs plot as below, where outliers are labeled with "+" in red

```
 n <- nrow(iris2)
 pch <- rep(".", n)
 pch[outliers] <- "+"
 col <- rep("black", n)
 col[outliers] <- "red"
 pairs(iris2, pch=pch, col=col)
```

4. Associations Rules

This section includes association rule mining, pruning redundant rules, and visualising association rules.

Association rule mining

```
# Association Rule Mining:
# Following examples use The Titanic dataset, a 4-dimensional table
# with summarized information  on the fate of passengers on the
# Titanic according to social class, sex, age and survival
# It can be found in https://www.rdatamining.com/datasets


 # get current script folder
 myPath <- dirname(rstudioapi::getSourceEditorContext()$path)

 #load dataset (assuming it is in script's folder)
 load(paste0(myPath,"/titanic.raw.rdata"))
 str(Titanic)
```

use APRIORI algorithm for association rule mining [Agrawal and Srikant, 1994]. package arules [Hahsler et al., 2014] implements it in ***apriori()*** function

```
library(arules)
 # find association rules with default settings
 rules <- apriori(titanic.raw)
 inspect(rules)
 ## use code below if above code does not work
 arules::inspect(rules)
```

```r
# rules with rhs (right-hand side) 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"))

rules.sorted <- sort(rules, by="lift")
inspect(rules.sorted)

# Removing Redundancy, find redundant rules
redundant <- is.redundant(rules.sorted)
which(redundant)
# remove redundant rules
rules.pruned <- rules.sorted[!redundant]
inspect(rules.pruned)

# Visualizing Association Rules
library(arulesViz)
plot(rules.pruned[1:3])
plot(rules.pruned[1:3], method="graph", control=list(type="items"))
plot(rules.pruned[1:3], method="paracoord", control=list(reorder=TRUE))
```