Introduction to Data Mining with R and Data Import/Export in R[1]

Yanchang Zhao

http://www.RDataMining.com

30 September 2014

[1] Presented at UJAT in Sept 2014

Questions

- Do you know data mining and its algorithms and techniques?

Questions

- Do you know data mining and its algorithms and techniques?
- Have you heard of R?

Questions

- Do you know data mining and its algorithms and techniques?
- Have you heard of R?
- Have you used R in your research or projects?

Outline

Introduction to R

R Packages and Functions for Data Mining

Data Import and Export

Online Resources

What is R?

- R [2] is a free software environment for statistical computing and graphics.
- R can be easily extended with 5,800+ packages available on CRAN[3] (as of 13 Sept 2014).
- Many other packages provided on Bioconductor[4], R-Forge[5], GitHub[6], etc.
- R manuals on CRAN[7]
  - *An Introduction to R*
  - *The R Language Definition*
  - *R Data Import/Export*
  - …

[2] `http://www.r-project.org/`

[3] `http://cran.r-project.org/`

[4] `http://www.bioconductor.org/`

[5] `http://r-forge.r-project.org/`

[6] `https://github.com/`

[7] `http://cran.r-project.org/manuals.html`

Why R?

- R is widely used in both academia and **industry**.
- R was ranked no. 1 in the KDnuggets 2014 poll on *Top Languages for analytics, data mining, data science*[8] (actually R has been no. 1 in 2011, 2012 & 2013!).
- The CRAN Task Views [9] provide collections of packages for different tasks.
  - Machine learning & atatistical learning
  - Cluster analysis & finite mixture models
  - Time series analysis
  - Multivariate statistics
  - Analysis of spatial data
  - ...

[8] `http://www.kdnuggets.com/polls/2014/languages-analytics-data-mining-data-science.html`

[9] `http://cran.r-project.org/web/views/`

Outline

Introduction to R

R Packages and Functions for Data Mining

Data Import and Export

Online Resources

Classification with R

- Decision trees: *rpart*, *party*
- Random forest: *randomForest*, *party*
- SVM: *e1071*, *kernlab*
- Neural networks: *nnet*, *neuralnet*, *RSNNS*
- Performance evaluation: *ROCR*

Clustering with R

- *k*-means: *kmeans()*, *kmeansruns()*[10]
- *k*-medoids: *pam()*, *pamk()*
- Hierarchical clustering: *hclust()*, *agnes()*, *diana()*
- DBSCAN: *fpc*
- BIRCH: *birch*

[10] Functions are followed with "()", and others are packages.

Association Rule Mining with R

- Association rules: *apriori()*, *eclat()* in package *arules*
- Sequential patterns: *arulesSequence*
- Visualisation of associations: *arulesViz*

Text Mining with R

- Text mining: *tm*
- Topic modelling: *topicmodels*, *lda*
- Word cloud: *wordcloud*
- Twitter data access: *twitteR*

Time Series Analysis with R

- Time series decomposition: *decomp(), decompose(), arima(), stl()*
- Time series forecasting: *forecast*
- Time Series Clustering: *TSclust*
- Dynamic Time Warping (DTW): *dtw*

Social Network Analysis with R

- Packages: *igraph*, *sna*
- Centrality measures: *degree(), betweenness(), closeness(), transitivity()*
- Clusters: *clusters(), no.clusters()*
- Cliques: *cliques(), largest.cliques(), maximal.cliques(), clique.number()*
- Community detection: *fastgreedy.community(), spinglass.community()*

R and Big Data

- Hadoop
  - Hadoop (or YARN) - a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models
  - R Packages: *RHadoop*, *RHIPE*
- Spark
  - Spark - a fast and general engine for large-scale data processing, which can be 100 times faster than Hadoop
  - *SparkR* - R frontend for Spark
- H2O
  - H2O - an open source in-memory prediction engine for big data science
  - R Package: *h2o*
- MongoDB
  - MongoDB - an open-source document database
  - R packages: *rmongodb*, *RMongo*

R and Hadoop

- Packages: *RHadoop*, *RHive*
- RHadoop[11] is a collection of R packages:
  - *rmr2* - perform data analysis with R via MapReduce on a Hadoop cluster
  - *rhdfs* - connect to Hadoop Distributed File System (HDFS)
  - *rhbase* - connect to the NoSQL HBase database
  - ...
- You can play with it on a single PC (in standalone or pseudo-distributed mode), and your code developed on that will be able to work on a cluster of PCs (in full-distributed mode)!
- Step-by-Step Guide to Setting Up an R-Hadoop System `http://www.rdatamining.com/big-data/r-hadoop-setup-guide`

[11] `https://github.com/RevolutionAnalytics/RHadoop/wiki`

Outline

Introduction to R

R Packages and Functions for Data Mining

Data Import and Export

Online Resources

Data Import and Export [12]

Read data from and write data to

- R native formats (incl. `Rdata` and `RDS`)
- CSV files
- EXCEL files
- ODBC databases
- SAS databases

R Data Import/Export:

- `http://cran.r-project.org/doc/manuals/R-data.pdf`

[12] Chapter 2: Data Import and Export, in book *R and Data Mining: Examples and Case Studies*. `http://www.rdatamining.com/docs/RDataMining.pdf`

Save and Load R Objects

- `save()`: save R objects into a `.Rdata` file
- `load()`: read R objects from a `.Rdata` file
- `rm()`: remove objects from R

```
a <- 1:10
save(a, file = "./data/dumData.Rdata")
rm(a)
a

## Error:  object 'a' not found

load("./data/dumData.Rdata")
a

##  [1]  1  2  3  4  5  6  7  8  9 10
```

Save and Load R Objects - More Functions

- `save.image()`:
  save current workspace to a file
  It saves everything!
- `readRDS()`:
  read a single R object from a `.rds` file
- `saveRDS()`:
  save a single R object to a file
- Advantage of `readRDS()` and `saveRDS()`:
  You can restore the data under a different object name.
- Advantage of `load()` and `save()`:
  You can save multiple R objects to one file.

Import from and Export to .CSV Files

- `write.csv()`: write an R object to a `.CSV` file
- `read.csv()`: read an R object from a `.CSV` file

```
# create a data frame
var1 <- 1:5
var2 <- (1:5)/10
var3 <- c("R", "and", "Data Mining", "Examples", "Case Studies")
df1 <- data.frame(var1, var2, var3)
names(df1) <- c("VarInt", "VarReal", "VarChar")
# save to a csv file
write.csv(df1, "./data/dummmyData.csv", row.names = FALSE)
# read from a csv file
df2 <- read.csv("./data/dummmyData.csv")
print(df2)

##   VarInt VarReal      VarChar
## 1      1     0.1            R
## 2      2     0.2          and
## 3      3     0.3  Data Mining
## 4      4     0.4     Examples
## 5      5     0.5 Case Studies
```

Import from and Export to EXCEL Files

Package *xlsx*: read, write, format Excel 2007 and Excel 97/2000/XP/2003 files

```
library(xlsx)
xlsx.file <- "./data/dummmyData.xlsx"
write.xlsx(df2, xlsx.file, sheetName = "sheet1", row.names = F)
df3 <- read.xlsx(xlsx.file, sheetName = "sheet1")
df3

##   VarInt VarReal      VarChar
## 1      1     0.1            R
## 2      2     0.2          and
## 3      3     0.3  Data Mining
## 4      4     0.4     Examples
## 5      5     0.5 Case Studies
```

Read from Databases

- Package *RODBC*: provides connection to ODBC databases.
- Function `odbcConnect()`: sets up a connection to database
- `sqlQuery()`: sends an SQL query to the database
- `odbcClose()` closes the connection.

```
library(RODBC)
db <- odbcConnect(dsn = "servername", uid = "userid",
                  pwd = "******")
sql <- "SELECT * FROM lib.table WHERE ..."
# or read query from file
sql <- readChar("myQuery.sql", nchars=99999)
myData <- sqlQuery(db, sql, errors=TRUE)
odbcClose(db)
```

Read from Databases

- Package *RODBC*: provides connection to ODBC databases.
- Function `odbcConnect()`: sets up a connection to database
- `sqlQuery()`: sends an SQL query to the database
- `odbcClose()` closes the connection.

```r
library(RODBC)
db <- odbcConnect(dsn = "servername", uid = "userid",
                  pwd = "******")
sql <- "SELECT * FROM lib.table WHERE ..."
# or read query from file
sql <- readChar("myQuery.sql", nchars=99999)
myData <- sqlQuery(db, sql, errors=TRUE)
odbcClose(db)
```

Functions `sqlFetch()`, `sqlSave()` and `sqlUpdate()`: read, write or update a table in an ODBC database

Import Data from SAS

Package *foreign* provides function `read.ssd()` for importing SAS datasets (`.sas7bdat` files) into R.

```r
library(foreign) # for importing SAS data
# the path of SAS on your computer
sashome <- "C:/Program Files/SAS/SASFoundation/9.2"
filepath <- "./data"
# filename should be no more than 8 characters, without extension
fileName <- "dumData"
# read data from a SAS dataset
a <- read.ssd(file.path(filepath), fileName,
              sascmd=file.path(sashome, "sas.exe"))
```

Import Data from SAS

Package *foreign* provides function `read.ssd()` for importing SAS datasets (`.sas7bdat` files) into R.

```r
library(foreign) # for importing SAS data
# the path of SAS on your computer
sashome <- "C:/Program Files/SAS/SASFoundation/9.2"
filepath <- "./data"
# filename should be no more than 8 characters, without extension
fileName <- "dumData"
# read data from a SAS dataset
a <- read.ssd(file.path(filepath), fileName,
              sascmd=file.path(sashome, "sas.exe"))
```

Another way: using function `read.xport()` to read a file in SAS Transport (XPORT) format

Outline

Introduction to R

R Packages and Functions for Data Mining

Data Import and Export

Online Resources

Online Resources

- RDataMining website

  `http://www.rdatamining.com`

  - R Reference Card for Data Mining
  - R and Data Mining: Examples and Case Studies
- RDataMining Group on LinkedIn (7,000+ members)

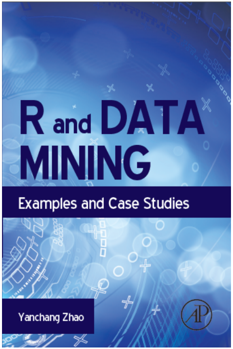
  `http://group.rdatamining.com`
- RDataMining on Twitter (1,700+ followers)

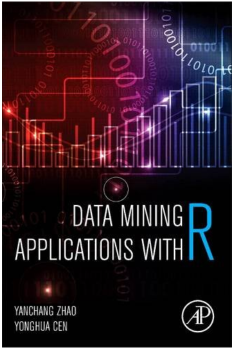
  `@RDataMining`
- Free online courses

  `http://www.rdatamining.com/resources/courses`
- Online documents

  `http://www.rdatamining.com/resources/onlinedocs`

The End





Thanks!

Email: yanchang(at)rdatamining.com