Data Pre-Processing in R

L. Torgo

ltorgo@fc.up.pt
Faculdade de Ciências / LIAAD-INESC TEC, LA
Universidade do Porto

Nov, 2014

What is Data Pre-Processing?

Data Pre-Processing

Set of steps that may be necessary to carry out before any further analysis takes place on the available data

Some Motivations for Data Pre-Processing

- Several data mining methods are sensitive to the scale and/or type of the variables
  - Different variables (columns of our data sets) may have rather different scales
  - Some methods are not able to handle either nominal or numeric variables
- We may need to “create” new variables to achieve our objectives
  - Sometimes we are more interested in relative values (variations) than absolute values
  - We may be aware of some domain-specific mathematical relationship among two or more variables that is important for the task
- Frequently we have data sets with unknown variable values
- Our data set may be too large for some methods to be applicable

Normalizing Numeric Variables

Goal

Make all variables have the same scale - usually a scale where all have mean 0 and standard deviation 1

$$y = \frac{x - \bar{x}}{\sigma_x}$$

```
data(algae,package="DMwR")
norm.algae <- cbind(algae[,1:3],
                    scale(algae[,-c(1:3,12:18)]),
                    algae[,12:18])
```

Working with relative values instead of absolute values

Why?

Frequent technique that is used in time series analysis to avoid trend effects

$$y_i = \frac{x_i - x_{i-1}}{x_{i-1}}$$

```
x <- rnorm(100, mean = 100, sd = 3)
head(x)

## [1] 106.86 102.45 102.37  99.12  93.05  97.90

vx <- diff(x)/x[-length(x)]
head(vx)

## [1] -0.0412998 -0.0007987 -0.0317119 -0.0612026  0.0520758  0.0004876
```

An example with real-world time series data

The S&P 500 stock market index

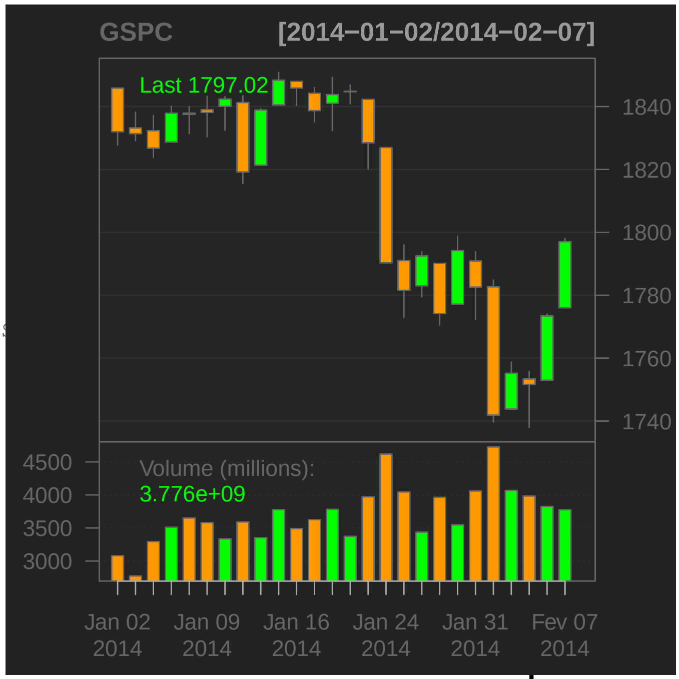
```
library(quantmod)  # extra package
getSymbols("^GSPC", from = "2014-01-01")

## [1] "GSPC"

head(GSPC, 3)

##            GSPC.Open GSPC.High GSPC.Low GS
## 2014-01-02      1846      1846     1828
## 2014-01-03      1833      1838     1829
## 2014-01-06      1832      1837     1824
##            GSPC.Adjusted
## 2014-01-02          1832
## 2014-01-03          1831
## 2014-01-06          1827
```

```
candleChart(GSPC)
```



An example with real-world time series data (2)

The S&P 500 stock market index

```
head(Cl(GSPC))

##            GSPC.Close
## 2014-01-02       1832
## 2014-01-03       1831
## 2014-01-06       1827
## 2014-01-07       1838
## 2014-01-08       1837
## 2014-01-09       1838

head(Delt(Cl(GSPC)))

##            Delt.1.arithmetic
## 2014-01-02                NA
## 2014-01-03        -0.0003330
## 2014-01-06        -0.0025118
## 2014-01-07         0.0060818
## 2014-01-08        -0.0002122
## 2014-01-09         0.0003483
```

Dealing with Unknown Values

Some Possible Strategies

- Remove all lines in a data set with some unknown value
- Fill-in the unknowns with the most common value (a statistic of centrality)
- Fill-in with the most common value on the cases that are more “similar” to the one with unknowns
- Explore eventual correlations between variables
- etc.

Some illustrations in R

```
library(DMwR)
data(algae)
head(algae[!complete.cases(algae),],3)

##    season  size speed mxPH mnO2   Cl  NO3 NH4 oPO4 PO4 Chla   a1  a2 a3 a4
## 28 autumn small  high  6.8 11.1 9.00 0.63  20  4.0  NA  2.7 30.3 1.9  0  0
## 38 spring small  high  8.0   NA 1.45 0.81  10  2.5   3  0.3 75.8 0.0  0  0
## 48 winter small   low   NA 12.6 9.00 0.23  10  5.0   6  1.1 35.5 0.0  0  0
##     a5  a6  a7
## 28 2.1 1.4 2.1
## 38 0.0 0.0 0.0
## 48 0.0 0.0 0.0

nrow(algae[!complete.cases(algae),])

## [1] 16

noNA.algae <- na.omit(algae)
```

Some illustrations in R (2)

```
noNA.algae <- centralImputation(algae)
nrow(noNA.algae[!complete.cases(noNA.algae),])

## [1] 0

noNA.algae <- knnImputation(algae,k=10)
nrow(noNA.algae[!complete.cases(noNA.algae),])

## [1] 0
```

Reducing the dimension of the data set

Motivations

- Some data mining methods may be unable to handle very large data sets
- The computation time to obtain a certain model may be too large for the application
- We may want simpler models
- etc.

Some strategies

- Reduce the number of variables
- Reduce the number of cases
- Reduce the number of values on the variables

Reducing the number of cases

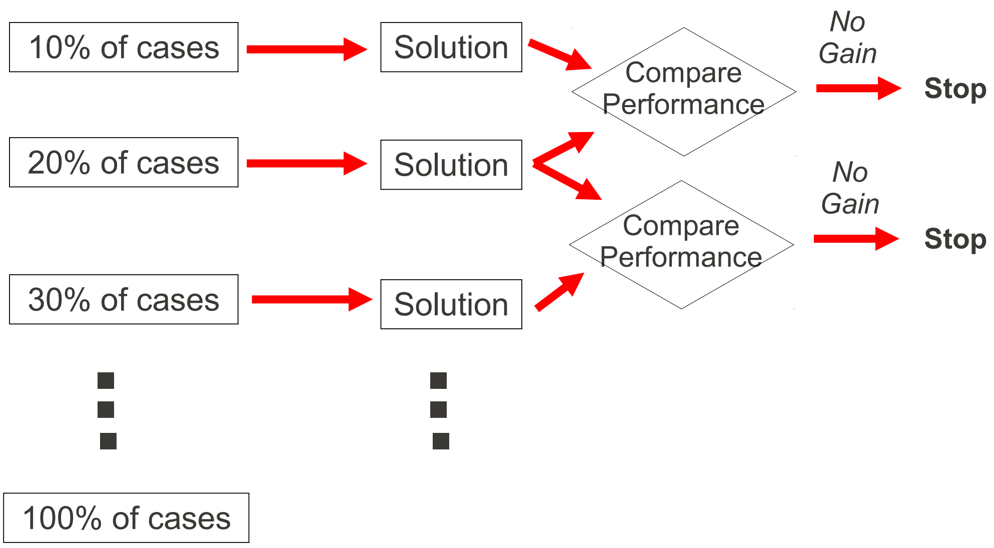
Resampling strategies

Reducing the number of cases usually is carried out through some form of random resampling of the original data

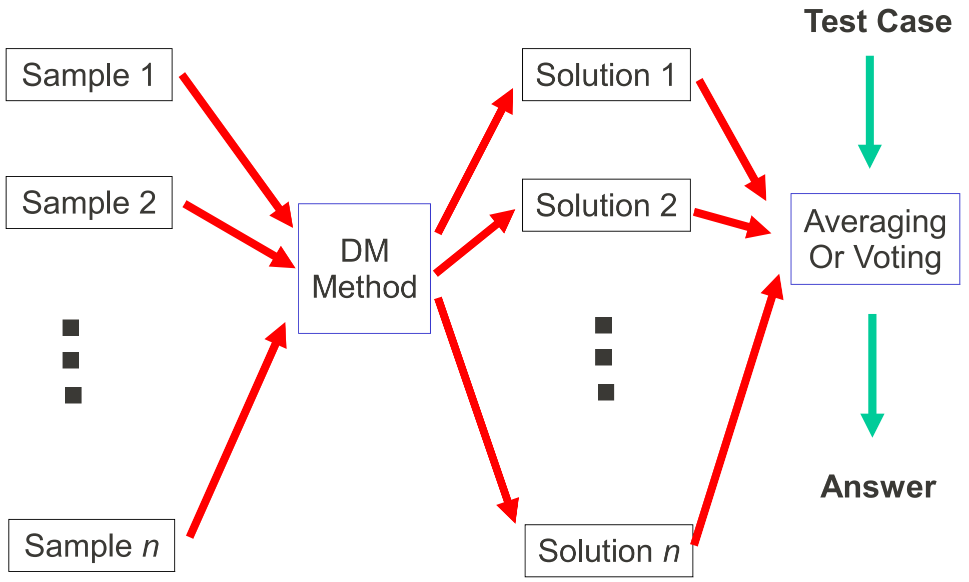
Some possible methods:

- Random selection of a sub-set of the data set
- Random and stratified selection of a sub-set of the data
- Incremental sampling
- Multiple sample and/or models

Incremental Sampling



Multiple Samples and/or Models



Reducing the number of values in numeric variables

Main motivation: Some techniques have their computational complexity heavily dependent on the number of values of the numeric variables.
A few simple techniques that may help on these situations:

- Rounding
- Values discretization
  - Grouping values
    - Equal-size groups
    - Equal-frequency groups
    - k-means method
    - etc.

Some illustrations with R

Random samples of a data set. Peeking 70% of the rows of one data set:

```
data(Boston, package = "MASS")
idx <- sample(1:nrow(Boston), as.integer(0.7 * nrow(Boston)))
smpl <- Boston[idx, ]
rmng <- Boston[-idx, ]
nrow(smpl)

## [1] 354

nrow(rmng)

## [1] 152
```

Some illustrations with R (2)

Discretizing a variable into 4 intervals.

- Equal-width

```
Boston$age <- cut(Boston$age, 4)
table(Boston$age)

##
##   (2.8,27.1] (27.1,51.5] (51.5,75.8]  (75.8,100]
##           51          97          96         262
```

- Equal-frequency

```
data(Boston,package='MASS')
Boston$age <- cut(Boston$age,
                  quantile(Boston$age,probs=seq(0,1,.25)))
table(Boston$age)

##
##     (2.9,45]   (45,77.5] (77.5,94.1]  (94.1,100]
##          126         126         126         127
```

Hands on Data Pre-Processing - the Algae data set

Using the Algae data set from package `DMwR` answer the following questions:

1. Try different methods for filling-in the missing values in the variable `Chla`, and compare them in terms of the resulting distribution of the variable.
2. Create a graph to compare the distributions obtained in the previous exercise.
3. Create a new variable for the Algae data set that results from discretizing the values of `a1` into three categories: `normal`, `high` and `extreme`. Select appropriate thresholds for these categories.