Modeling Data with Dependencies

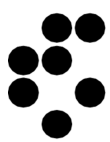
Solutions to Hands on Exercises

L. Torgo

ltorgo@fc.up.pt

Faculdade de Ciências / LIAAD-INESC TEC, LA
Universidade do Porto

Jan, 2017



Jožef Stefan Institute

Hands On Time Series

Package **quantmod** (an extra package that you need to install) contains several facilities to handle financial time series. Among them, the function `getSymbols` allows you to download the prices of financial assets from *yahoo finance*. Explore the help page of the function to try to understand how it works, and the answer the following:

1. Obtain the prices of Apple during the last year solution
2. Using these prices create a time series of the percentage variation of the Closing prices (tip: check function `Cl()` and `Delt` from package **quantmod**) solution
3. Create and embed data set of the previous series using function `createEmbedDS()` of package **DMwR2** solution
4. Split the data set in two consecutive periods. Train a random forest with the first and apply it to the second. solution
5. Analyse the results solution

Jožef Stefan Institute

Solution to Exercise 1

- Obtain the prices of Apple during the last year

```
library(quantmod)
library(lubridate)
getSymbols("AAPL",from=Sys.Date() - years(1))

## [1] "AAPL"

# checking
start(AAPL)

## [1] "2016-01-20"

end(AAPL)

## [1] "2017-01-19"
```

Go Back

Jožef Stefan Institute

Solution to Exercise 2

- Using these prices create a time series ...

```
apl <- Delt(Cl(AAPL))
```

Go Back

Jožef Stefan Institute

Solution to Exercise 3

- Create and embed data set ...

```
library(DMwR2)
library(TTR)
dat <- createEmbedDS(apl, emb=7)
dat <- data.frame(cbind(lag(apl,-1),
                        dat,
                        MA10=SMA(apl,10),
                        RSI=RSI(apl),
                        BB=BBands(apl)$pctB))
colnames(dat)[1] <- "FutureT"
dat <- na.omit(dat)
```

Go Back

Jožef Stefan Institute

Solution to Exercise 4

- Split the data set in two consecutive periods. Train a random forest with the first and apply it to the second.

```
set.seed(1234)
sztr <- as.integer(0.7*nrow(dat))
tr <- dat[1:sztr,]
ts <- dat[(sztr+1):nrow(dat),]
library(randomForest)
mdl <- randomForest(FutureT ~ ., tr, ntrees=1000)
preds <- predict(mdl, ts)
```

Go Back

Solution to Exercise 5

- Analyse the results

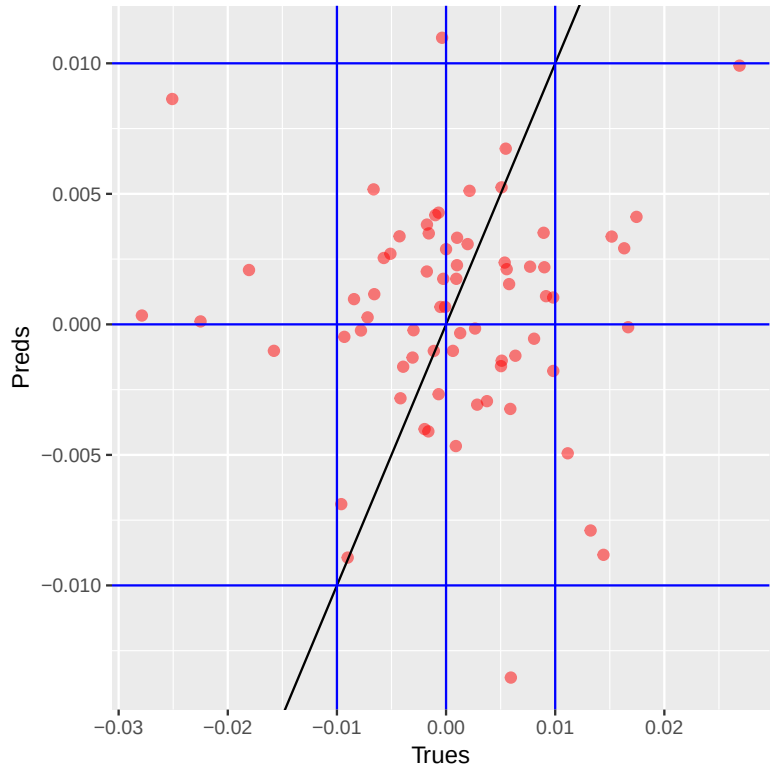
```
(mae <- mean(abs(preds-ts$FutureT)))

## [1] 0.007684985
```

Go Back

Jožef Stefan Institute

Solution to Exercise 5 - cont.

```
library(ggplot2)
dt <- data.frame(Trues=ts$FutureT, Preds=preds)
ggplot(dt, aes(x=Trues, y=Preds)) +
    geom_point(col="red", size=2, alpha=0.5) +
    geom_abline(slope=1,intercept=0) +
    geom_hline(yintercept=c(-0.01,0,0.01),col="blue") +
    geom_vline(xintercept=c(-0.01,0,0.01),col="blue") +
    guides(col=FALSE)
```



Jožef Stefan Institute

Hands On Spatial Forecasting

Using the `meuse` data set from package **sp**:

1. Build a multiple regression data set to predict the variable *cadmium* through the function `getVars()` shown during the classes. Explore other statistics apart from the defaults of the function. solution
2. Split the obtained data set randomly in two 70-30% partitions solution
3. Obtain an SVM with the larger partition solution
4. Apply the model to obtain predictions for the smaller partition and analyse the results solution

Jožef Stefan Institute

Solutions to Exercise 1

- Build a multiple regression data set ...

```
library(sp)
data(meuse)
coordinates(meuse) <- ~x+y
proj4string(meuse) <- CRS("+init=epsg:28992")
dat <- NULL
for(r in 1:nrow(meuse))
    dat <- rbind(dat,
                 getVars(meuse[r,],meuse[-r,],"cadmium",
                         nns=c(2,5,7),
                         funcs=c("mean","sd","median","max","min")))
dat <- data.frame(dat,tgtCad=meuse[["cadmium"]])
```

Go Back

Jožef Stefan Institute

Solutions to Exercise 2

- Split the obtained data set randomly in two 70-30% partitions

```
set.seed(1234)
idx <- sample(1:nrow(dat),as.integer(0.7*nrow(dat)))
tr <- dat[idx,]
ts <- dat[-idx,]
```

Go Back

Jožef Stefan Institute

Solutions to Exercise 3

- Obtain an SVM with the larger partition

```
library(e1071)

mdl <- svm(tgtCad ~ ., tr, cost=10, epsilon=0.01)
```

Go Back

Jožef Stefan Institute

Solutions to Exercise 4

- Apply the model to obtain predictions for the smaller partition and analyse the results

```
preds <- predict(mdl, ts)
mae <- mean(abs(preds-ts$tgtCad))
mae
```

```
## [1] 2.24393
```

```
library(DMwR)
regr.eval(ts$tgtCad, preds, train.y=tr$tgtCad)
```

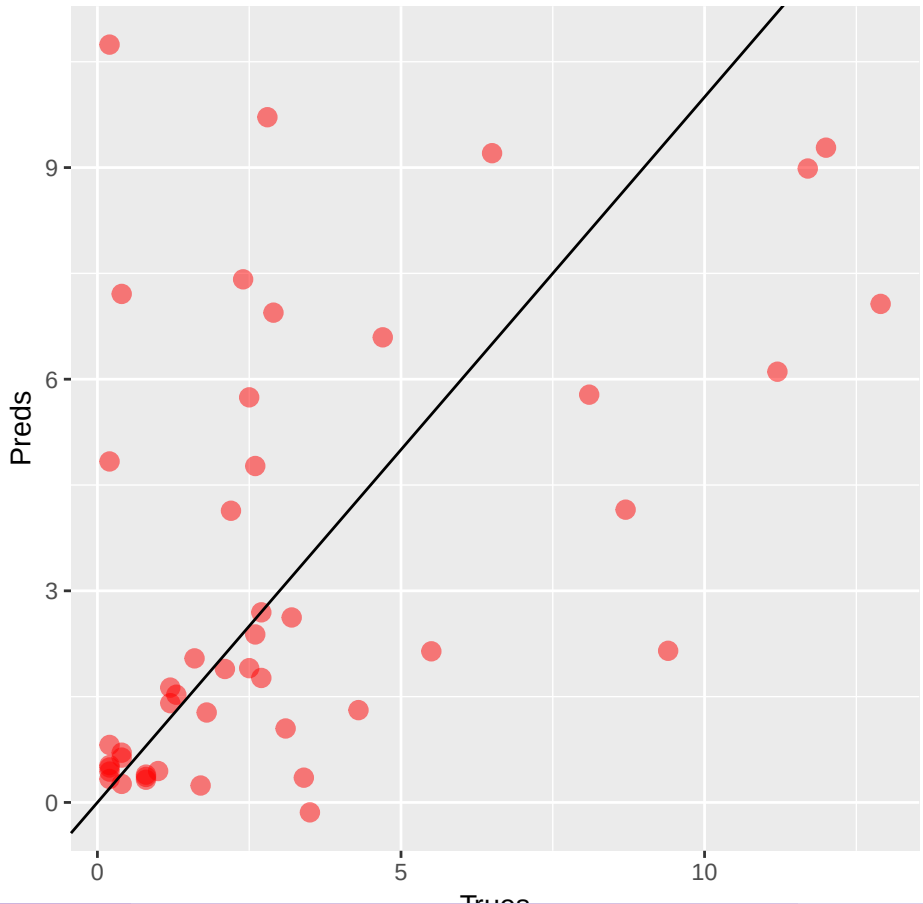
```
##         mae         mse        rmse        mape        nmse        nmae
##   2.2439297 10.7390299  3.2770459  2.6506198  0.9005765  0.8731525
```

Go Back

Jožef Stefan Institute

Solutions to Exercise 4 - cont.

```
library(ggplot2)
dt <- data.frame(Trues=ts$tgtCad, Preds=preds)
ggplot(dt, aes(x=Trues, y=Preds)) +
    geom_point( col="red", size=3, alpha=0.5) +
    geom_abline(slope=1,intercept=0) +
    guides(col=FALSE)
```



Jožef Stefan Institute