Regression exercises

The data files used in the exercises below can be found on the blackboard site! Download them from the blackboard site and save them in an appropriate folder on your computer or network drive. Point your R session to this location with this command setwd(D:/Your/Folder/Goes\_here).

# Exercise 1

Data from 24 chemical solutions have been collected in order to examine the association between the toxicity of the solution on the one hand and three explanatory variables on the other hand. The data are saved in the files lser.xlsx with the following variables:

* tox : toxicity of the dissolution
* base : ablility to accept hydrogen ions
* acid : ablility to liberate hydrogen ions
* colour : ability to change colour

1. The first step will be to load the data into R. The data file is given as an Excel file which means we need an extra package to read the xlsx file. There are several packages that can do this, but we wil use the package XLCOnnect. If not installed before, do this using this command:

* > install.packages("XLConnect")
* Load the package and read in the excel file.
* > require(XLConnect)  
  > theData <- readWorksheet(loadWorkbook("lser.xlsx"), sheet=1)

1. This is a small dataset, so we can visualise it easily. How a look at the data by creating a matrix scatter plot.

* > pairs(theData)
* Describe the relations between the explanatory variables and the response variable.

1. Fit a multiple linear regression model with tox as response and base, acid and colour as explanatory variables. Give the fitted model (so with the estimates for the regression coefficients). What is the interpretation of the parameter estimates? We have not discussed hypothesis tests yet, but can you figure out already which predictors are important and which ones are not?

* > model <- lm(tox ~ colour + base + acid, data=theData)  
  > summary(model)

1. Calculate the expected toxicity for a solvent which has base=0.60, acid=0.95, and colour=0.52.

* > predict(model, data.frame(colour=0.52, base=0.60, acid=0.95) )

1. Create a scatter plot of observed values on the x-axis and predicted values on the y-axis. Draw a diagonal line to emphasize perfect predictions.

* > ypred <- predict(model)  
  > plot(theData$tox, ypred)  
  > abline(a=0, b=1)
* What is your impression of the model after this visualisation?

# Exercise 2

The data originates from 1H-NMR analysis of 40 table wines of different origin and color. No buffer and/or pH adjustment has been used prior to analysis therefore the spectra present a quite strong misalignment of the NMR resonance signals.

The data set contains a data frame where the first 17 columns are measured chemical values (the responses) and the other columns (8712) are NMR intensity values for the different chemical shift (in ppm). The column names either give the chemical or the chemical shift.

Referece for data set: F.H. Larsen, F. van den Berg, S.B. Engelsen, An exploratory chemometric study of 1H-NMR spectra of table wines. J.Chemom. 20 (2006) 198-208

1. Load in the data file "winedata.csv".

* > data <- as.matrix(read.csv("winedata.csv", header=TRUE, row.names=1))
* Note the code to read the file. Adaptions are made since he first columns in the data files contains the row names and our ridge and lasso functions require a matrix instead of a data.frame.

1. Inspect the data. How many columns and how many objects are there? What do they represent?

* > dim(data)
* > colnames(data)  
  > rownames(data)  
  > colnames(data[,1:17])

1. We will be using the ridge regression function from the package GLMNET. If this package was not installed before, install it using this command:

* > install.packages("glmnet")
* The next step would be to load the package using this command:
* > library(glmnet)
* After loading it, we create 2 variables x and y for the dependant and independent variables.
* > x=data[, -c(1:17)] #First 17 columns are response variables  
  > y=data[, 1]

1. For ridge regression we need to optimise the value for lambda. We want to search lambda in the range 1e10 and 1e-2. Create a 100-point grid between these 2 numbers for lambda.

* > grid <- 10^seq(10,-10,length=200)
* Inspect the variable grid.

1. For finding the optimal lambda value we use the function cv.glmnet. Have a look at the help page and see what the parameters for this function are. Note parameter alpha: this parameter switches between ridge regression (alpha=0) and the lasso (alpha=1).

* > ?cv.glmnet

1. Run cv.glmnet with all chemical shifts as predictors, trying to predict the first chemical malicAcid. Plot the results from the cross-validation and give the optimal value for lambda from the plot.

* > mdl <- cv.glmnet(x=x, y=y, alpha=0, lambda=grid)  
  > plot(mdl)

1. Using the optimal lambda, get predicted values for the response. Create a scatter plot of observed values versus the predicted values for the trait. Add the line y=x to the plot. What does this line represent?

* > ypred <- predict(mdl, newx=x) #get predictions for x  
  > plot(y, ypred)  
  > abline(a=0, b=1)

1. Calculate the explained variance R2 for this model.

* > sum((ypred-mean(y))^2)/sum((y-mean(y))^2)

1. Repeat the ridge regression models for some of the other response variables (just pick a few from the 17 available ones). From your choice, what model has the highest R2? How is this reflected in the scatterplots?
2. Ridge regression is not selecting variables. For this we can use the lasso. Build a lasso regression model for predicting the first chemical malicAcid, using the optimal lambda. Calculate R2 values for this models and compare them with the ridge regression model of question f. Also create a scatterplot with the predicted versus observed values.

* > x=data[, -c(1:17)] #First 17 columns are response variables  
  > y=data[, 1]  
  >   
  > mdl2 <- cv.glmnet(x=x, y=y, alpha=1, lambda=grid)  
  > plot(mdl2)  
  > ypred2 <- predict(mdl2, newx=x)  
  > R2 <- sum((ypred2-mean(y))^2)/sum((y-mean(y))^2)  
  > plot(y, ypred2, main=sprintf('%s lasso R^2=%.2f', colnames(data)[1], R2))  
  > abline(a=0, b=1)
* What do you conclude?

1. Have a look at the selected variables in the final lasso regression model. Show the predictors that have non-zero regression coefficients.

* > coef(mdl2)
* > colnames(x)[which(coef(mdl2) != 0)]

1. For the same response variables you used in question (i), create lasso models and calculate R2 values. How do the lasso models compare to the ridge regression counterparts?

This time we will focus on cross validation of the models.

1. Use leave-one-out crossvalidation to assess the predictive properties of the lasso model predicting the first chemical malicAcid (see questions f and j). Set one observation aside and build the model on the remaining observations. Repeat this so that all observations have been set aside once. Using the predicted values for all observations, calculate the Q2 value and create a scatter plot of the predicted versus the observed values. What is your conclusion when you compare this Q2 with the R2 from exercise h and the scatter plot with the one from question j? Runnig the leave-one-out crossvalidation may take some time to compute!

* > x=data[, -c(1:17)] #First 17 columns are response variables  
  > y=data[, 1]  
  >   
  > ypred <- rep(NA, nrow(data))  
  > for (i in 1:nrow(data)) {  
  + mdl <- cv.glmnet(x=x[-i,], y=y[-i], alpha=1, lambda=grid)  
  + ypred[i] <- predict(mdl, newx=t(x[i,]))  
  + }  
  > Q2 <- sum((ypred-mean(y))^2)/sum((y-mean(y))^2)  
  > plot(y, ypred, main=sprintf('%s lasso Q^2=%.2f', colnames(data)[1], Q2))  
  > abline(a=0, b=1)
* This last exercise took a long time to compute which is a general problem with cross validation routines. To speed up calculations, leave-one-out crossvalidation is often replaced with leave-n-out (n=2,3,4...), decreasing the number of models calculated. Another strategy would be to create a fixed number of folds (say 10), make it a 10-fold crossvalidation. A useful R function for creating all the folds is the split function. Example code for 10 **random** splits of our data set would be split(sample(1:nrow(data)), 1:10).

1. Optional: when there is still time, program a 10-fold cross validation.