Compulsory Exercise 2: Group 37

TMA4268 Statistical Learning V2019

Anders Bendiksen and Helge Bergo

02 April, 2020

Problem 1 (10p)

a) Ridge Regression (2p)

Show that the ridge regression estimator is $\hat{\beta}_{Ridge} = (X^T X + \lambda I)^{-1} X^T y$.

b) (2p)

Find the expected value and the variance-covariance matrix of $\hat{\beta}_{Ridge}$ (1P each).

c) (2P) - Multiple choice

(i) TRUE
(ii) FALSE
(iii) FALSE
(iv) TRUE

d) Forward Selection

```
library(ISLR)
set.seed(1)
train.ind = sample(1:nrow(College), 0.5*nrow(College))
college.train = College[train.ind,]
college.test = College[-train.ind,]
```

After dividing the data into a training and test set, the `regsubsets` function was used to create a forward selection model on the data, from the `leaps`-library.

```
library(leaps)
regfit.fwd = regsubsets(Outstate~.,data=college.train,method="forward", nvmax = 18)
reg.summary = summary(regfit.fwd)
```

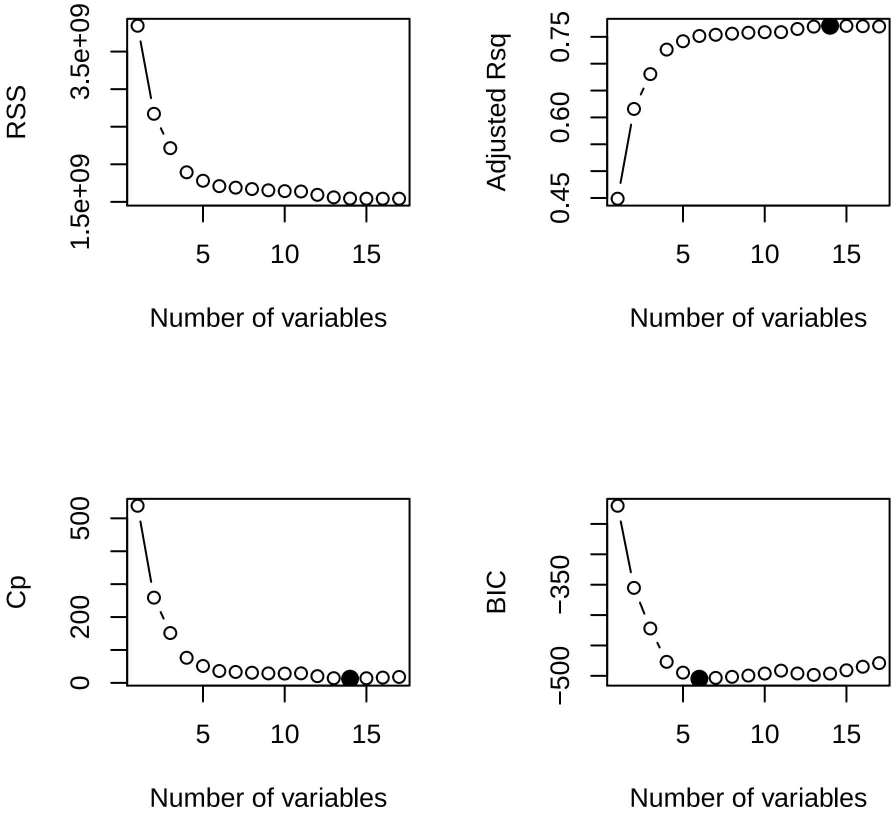
To decide on which model is best, the number of variables used in the selection was plotted against `RSS`, `Cp`, `BIC` and `adjusted R$^2$`.

```
par(mfrow=c(2,2))
plot(reg.summary$rss,xlab="Number of variables",ylab="RSS",type="b")
```

```r
plot(reg.summary$adjr2,xlab="Number of variables",ylab="Adjusted Rsq",type="b")
max.adjr2 = which.max(reg.summary$adjr2)
points(max.adjr2,reg.summary$adjr2[max.adjr2], col="black",cex=2,pch=20)

plot(reg.summary$cp,xlab="Number of variables",ylab="Cp",type="b")
min.cp = which.min(reg.summary$cp)
points(min.cp,reg.summary$cp[min.cp], col="black",cex=2,pch=20)

plot(reg.summary$bic,xlab="Number of variables",ylab="BIC",type="b")
min.bic = which.min(reg.summary$bic)
points(min.bic,reg.summary$bic[min.bic], col="black",cex=2,pch=20)
```



The maximum `adjusted` R^2 is the one with 14 variables, with a value of 0.7706887, shown as a filled dot in the upper right plot. This is also the same number of variables as for the lowest Cp. However, all the plots are pretty flat after around 6 or 7 variables used, and it seems like using only 6 variables still gives a good `adjusted` R^2 value of 0.7516133, without the increased complexity of adding 7 more variables. The model is then:

```r
coef(regfit.fwd,6)
```

```
##   (Intercept)    PrivateYes    Room.Board      Terminal   perc.alumni 
## -4726.8810613  2717.7019276     1.1032433    36.9990286    59.0863753 
##        Expend     Grad.Rate 
##     0.1930814    33.8303314
```

For the MSE, the following code calculates the MSE for all the variables.

```r
val.errors = rep(NA,17)
x.test = model.matrix(Outstate~.,data=college.test) # notice the -index!
```

```
for (i in 1:17) {
    coefi = coef(regfit.fwd,id=i)
    pred = x.test[,names(coefi)]%*%coefi
    val.errors[i] = mean((college.test$Outstate-pred)^2)
}

# plot(sqrt(val.errors),xlab="Number of variables", ylab="Root MSE",ylim=c(1500,5000) ,pch=19,type="b")
# points(sqrt(regfit.fwd$rss[-1]/180),col="blue",pch=19,type="b")
# legend("topright",legend=c("Training","Validation"),col=c("black","blue"),pch=19)
```

The MSE of the model with 6 variables is then:

```
val.errors[6]
```

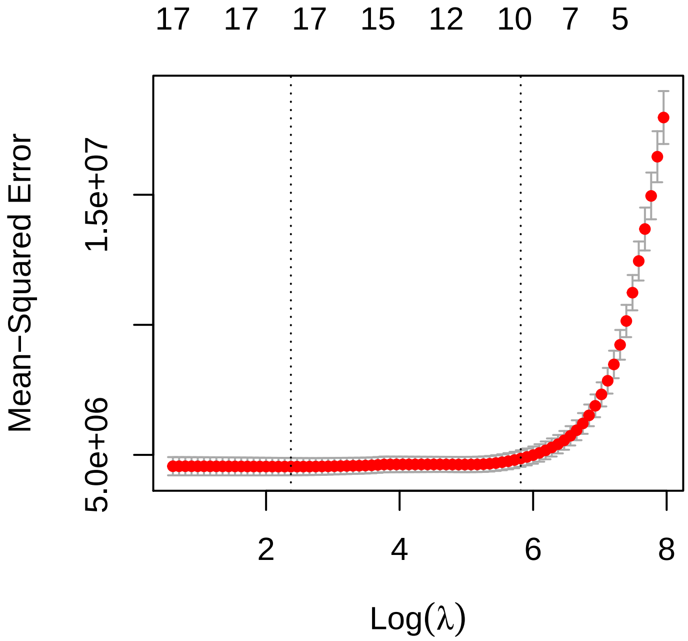
```
## [1] 3844857
```

e) (2p)

Using the Lasso method from the `glmnet`-library, a new model was selected.

To select the tuning parameter λ, cross-validation was performed, and the λ giving the lowest MSE was selected.

```
cv.out = cv.glmnet(x.train,y.train, alpha = 1)
plot(cv.out)
```



```
best.lambda = cv.out$lambda.min
best.lambda
```

```
## [1] 10.7207
```

This was used on the test set, to get the MSE for the

```
lasso.pred = predict(lasso.model,s=best.lambda ,newx=x.test)
MSE = mean((lasso.pred-y.test)^2)
MSE
```

```
## [1] 3688061
```

Finally, the coefficients of the model are shown here:

```r
lasso.coef = predict(cv.out,type="coefficients",s=best.lambda)[1:18,]
lasso.coef
```

```
##   (Intercept)    PrivateYes          Apps        Accept        Enroll 
## -1.172140e+03  2.230467e+03 -2.825215e-01  6.615811e-01 -3.778631e-01 
##     Top10perc     Top25perc   F.Undergrad   P.Undergrad    Room.Board 
##  4.589180e+01 -1.485674e+01 -5.800132e-02 -5.713770e-02  1.088115e+00 
##         Books      Personal           PhD      Terminal     S.F.Ratio 
## -9.185125e-01 -3.005419e-01  4.013410e+00  2.996744e+01 -6.936391e+01 
##   perc.alumni        Expend     Grad.Rate 
##  4.686967e+01  1.480013e-01  2.431539e+01
```

Problem 2 (9p)

a) (2p) - Multiple choice

Which of the following statements are true, which false?

(i) A regression spline of order 3 with 4 knots has 8 basis functions.

(ii) A regression spline with polynomials of degree $M-1$ has continuous derivatives up to order $M-2$, but not at the knots.

(iii) A natural cubic spline is linear beyond the boundary knots.

(iv) A smoothing spline is (a shrunken version of) a natural cubic spline with knots at the values of all data points x_i for $i = 1, \ldots, n$.

(v)

(vi)

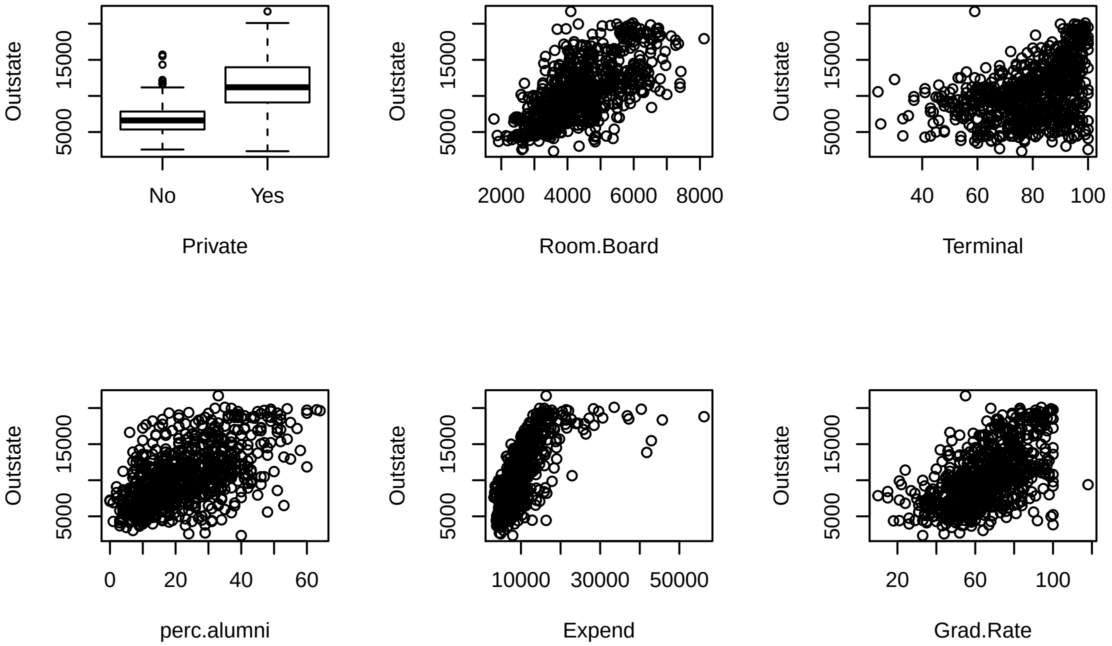
(vii)

(viii)

b) (2p)

Write down the basis functions for a cubic spline with knots at the quartiles q_1, q_2, q_3 of variable X.

c) (2p)

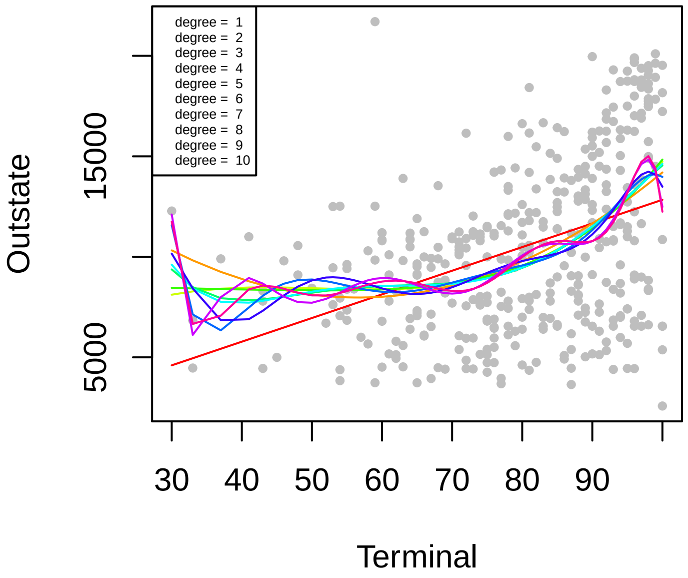


From these plots, it seems like `Room.board`, `perc.alumni` and `Grad.Rate` all have quite linear relationships with `Outstate`, while both `Terminal` and `Expend` seem to follow a non-linear relationship.

d) (3P)

(i) Fit polynomial regression models for `Outstate` with `Terminal` as the only covariate for a range of polynomial degrees ($d = 1, \ldots, 10$) and plot the results. Use the training data (`college.train`) for this task.

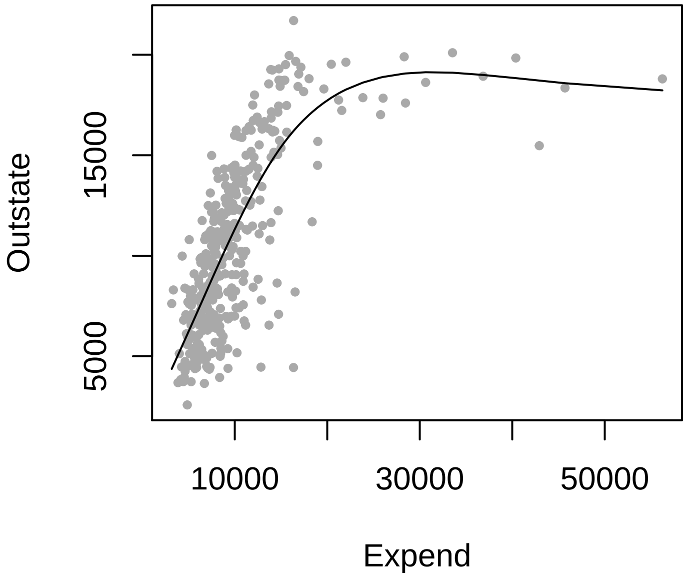
```
cols = rainbow(10)
deg = 1:10
polyfunc = function(d) {
  model = lm(Outstate ~ poly(Terminal,d), data=college.train)
  lines(cbind(college.train$Terminal, model$fit)[order(college.train$Terminal),],col=cols[d])
  pred = predict(model, college.train)
  mean((pred - college.train$Outstate)^2)
}
plot(college.train$Terminal, college.train$Outstate, col = "gray", pch=19,
     cex = 0.5, xlab = "Terminal", ylab = "Outstate")
MSE = sapply(deg, polyfunc)
legend("topleft",legend = paste("degree = ",deg), col = cols, cex = 0.4)
```



(ii) Still for the training data, choose a suitable smoothing spline model to predict `Outstate` as a function of `Expend` (again as the only covariate) and plot the fitted function into the scatterplot of `Outstate` against `Expend`. How did you choose the degrees of freedom?

```r
library(splines)
attach(college.train)
expend.range = range(Expend)
expend.grid = seq(from=expend.range[1],to=expend.range[2])

plot(Expend, Outstate, col = "darkgrey", pch=19, cex = 0.5)
fit.smoothspline = smooth.spline(Expend,Outstate,cv=TRUE)
lines(fit.smoothspline)
```



```r
# legend("bottomright",legend=paste("DF =",round(fit.smoothspline$df,2)),cex=.8)
```

The degrees of freedom was chosen using cross-validation, and the result was 4.661.

(iii) Report the corresponding training MSE for (i) and (ii). Did you expect that?

```
MSE.smoothspline.train = mean((predict(fit.smoothspline, college.train$Expend)$y -
                                 college.train$Outstate)^2)
MSE.smoothspline.train
```

```
## [1] 6871281
```

```
# MSE.smoothspline.test = mean((predict(fit.smoothspline, college.test$Expend)$y -
                                # college.test$Outstate)^2)
# MSE.smoothspline.test
MSE
```

```
##  [1] 15075161 14330586 14249448 14247330 14231485 14230392 14153207 14097911
##  [9] 13841526 13822205
```

The MSE for the polynomial regression is much higher than the MSE for the smoothing splines, but this makes a lot of sense when looking at the initial plots from 2.c). For the `Expend` variable, it seems like the data have a clearer trend than for the `Terminal` variable, and therefore the MSE is much lower.

Problem 3 (9p)

a) (2P) - Multiple choice

Which of the following statements are true, which false?

(i) Regression trees cannot handle categorical predictors.
(ii) Regression and classification trees are easy to interpret.
(iii) The random forest approaches improves bagging, because it reduces the variance of the predictor function by decorrelating the trees.
(iv) The number of trees B in bagging and random forests is a tuning parameter.

b) (4P)

Select one method from Module 8 (tree-based methods) in order to build a good model to predict `Outstate` in the `College` dataset that we used in problems 1 and 2. Explain your choice (pros/cons?) and how you chose the tuning parameter(s). Train the model using the training data and report the MSE for the test data.

c) (2p)

Compare the results (tests MSEs) among all the methods you used in Problems 1-3. Which method perform best in terms of prediction error? Which method would you choose if the aim is to develop an interpretable model?

Problem 4 (12P)

a) (2P) - Multiple choice

(i) TRUE
(ii) TRUE
(iii) TRUE
(iv) TRUE

b) (4P)

First, we convert the variables to factors, and fit a support vector classifier using the e1071 package and the svmfunction, and cross validation to find the best cost parameter.

```
d.train$diabetes <- as.factor(d.train$diabetes)
d.test$diabetes <- as.factor(d.test$diabetes)
library(e1071)
set.seed(1)

svm.linear = tune(svm,diabetes~.,data=d.train,kernel="linear",
                  ranges = list(cost=c(0.001,0.01,0.1,1,5,10,100)))
svm.linear.pred  = predict(svm.linear$best.model,d.test)
svm.linear.table = table(predict=svm.linear.pred, truth = d.test$diabetes)
svm.linear.error = sum(svm.linear.table[2:3]) / sum(svm.linear.table)
```

The confusion table and the misclassification error rate is:

```
svm.linear.table
```

```
##          truth
## predict    0   1
##        0 137  35
##        1  18  42
```

```
svm.linear.error
```

```
## [1] 0.2284483
```

Then, a support vector machine was fitted, again with cross validation, but this time to find the optimal combination of cost and γ.

```
svm.radial = tune(svm,diabetes~.,data=d.train,kernel="radial",
                       ranges = list(cost=c(0.001,0.01,0.1,1,5,10,100),
                                     gamma=c(0,0001, 0.001,0.01,0.1,1,5,10,100)))
svm.radial.pred  = predict(svm.radial$best.model,d.test)
svm.radial.table = table(predict=svm.radial.pred, truth = d.test$diabetes)
svm.radial.error = sum(svm.radial.table[2:3]) / sum(svm.radial.table)
```

The confusion table and the misclassification error rate is:

```
svm.radial.table
```

```
##          truth
## predict    0   1
##        0 140  38
##        1  15  39
```

```
svm.radial.error
```

```
## [1] 0.2284483
```

Comparing these two, the misclassification error rate is actually identical for the given test set, but there are some differences in the confusion matrices. There are more negative predictions in the radial model, both true and false negatives, 3 more on each. This is interesting, and shows the difference between the two types of boundaries and the impact a difference in cost and γ gives. For the given data, the preferred model would probably be the linear one, as this is both simpler, and gives a higher number of true positives. In the case of diabetes, misclassification in the form of false positives is better than false negatives, in our opinion.

c) (2P)

Comparing the SVMs to a linear discriminant analysis, the following code gives a fit using LDA.

```
lda.fit = lda(diabetes~., data = d.train)
lda.pred = predict(lda.fit,d.test)
lda.table = table(predict=lda.pred$class, truth = d.test$diabetes)
lda.error = sum(lda.table[2:3]) / sum(lda.table)
lda.table
```

```
##        truth
## predict   0   1
##       0 137  34
##       1  18  43
```

```
lda.error
```

```
## [1] 0.2241379
```

As can be seen, the misclassification rate is very similar, with only one less false negative compared to the support vector classifier. The main difference between the two methods is that the SVM uses only some observations as vectors to create the separating hyperplane, while the LDA uses all observations. This makes SVM less dependant on observations far from the hyperplane, while LDA is more affected by outliers in the data.

d) (2P) - Multiple choice

(i) FALSE
(ii) FALSE
(iii) TRUE
(iv) TRUE

e) (2P) Link to logistic regression and hinge loss.

Look at slides 71-73 of Module 9. Show that the loss function

$$\log(1+\exp(-y_i f(\boldsymbol{x}_i)))$$

is the deviance for the $y = -1, 1$ encoding in a logistic regression model.
Hint: $f(\boldsymbol{x}_i)$ corresponds to the linear predictor in the logistic regression approach.

Using $f(\boldsymbol{x}_i)$ as corresponding to the linear predictor in the logistic regression approach,

$$f(\boldsymbol{x})_i = \frac{e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_p x_{ip}}}{1+e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_p x_{ip}}}$$

the logistic regression model is on the form

$$p_i = \frac{e^{f(\boldsymbol{x}_i)}}{1+e^{f(\boldsymbol{x}_i)}}.$$

In logistic regression, the observations contribute by a weight $p_i(1-p_i)$, so the regression model can be rewritten to

$$f(\boldsymbol{x}_i) = \log(\frac{p_i}{1-p_i})$$

This means that the loss function

$$\log(1 + exp(-y_i f(\boldsymbol{x}_i))$$

is the deviance for the $y = -1, 1$ encoding in a logistic regression model.

Set er der jddjuddu

Problem 5 (10P)

The following dataset consists of 40 tissue samples with measurements of 1,000 genes. The first 20 tissues come from healthy patients and the remaining 20 come from a diseased patient group. The following code loads the dataset into your session with row names describing if the tissue comes from a diseased or healthy person.

```
# id <- "1VfVCQvWt121UN39NXZ4aR9Dmsbj-p9OU" # google file ID
# GeneData <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download", id),header=F)
# colnames(GeneData)[1:20] = paste(rep("H", 20), c(1:20), sep = "")
# colnames(GeneData)[21:40] = paste(rep("D", 20), c(1:20), sep = "")
# # row.names(GeneData) = paste(rep("G", 1000), c(1:1000), sep = "")
```

a) (2P)

Perform hierarchical clustering with complete, single and average linkage using **both** Euclidean distance and correlation-based distance on the dataset. Plot the dendograms. Hint: You can use `par(mfrow=c(1,3))` to plot all three dendograms on one line or `par(mfrow=c(2,3))` to plot all six together.

b) (2P)

Use these dendograms to cluster the tissues into two groups. Compare the groups with respect to the patient group the tissue comes from. Which linkage and distance measure performs best when we know the true state of the tissue?

c) (1P)

With Principal Component Analysis, the first principal component loading vector solves the following optimization problem,

$$\max_{\phi_{11},\ldots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \phi_{j1}^2 = 1.$$

Explain what ϕ, p, n and x are in this optimization problem and write down the formula for the first principal component scores.

d) (2P)

(i) (1P) Use PCA to plot the samples in two dimensions. Color the samples based on the tissues group of patients.
(ii) (1P) How much variance is explained by the first 5 PCs?

e) (1P)

Use your results from PCA to find which genes that vary the most across the two groups.

f) (2P)

Use K-means to separate the tissue samples into two groups. Plot the values in a two-dimensional space with PCA. What is the error rate of K-means?