Compulsory exercise 2: Group 24

TMA4268 Statistical Learning V2019

Silje Anfindsen and Clara Panchaud

18 mars, 2020

Problem 1

a)

Let's find the ridge regression estimator. Remember that $\hat{\beta}_{Ridge}$ minimizes $RSS + \lambda \sum_{j=1}^{p} \beta_j^2$. Let's rewrite this in matrix notation.

$$\begin{aligned}
min_{\beta}\{(y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta\} = & \quad \text{develop the expression} \\
min_{\beta}\{y^Ty - 2\beta^TX^Ty + \beta^TX^TX\beta + \lambda\beta^T\beta\} & \quad \text{take the derivative with respect to beta and set equal to 0} \\
-2X^Ty + 2X^TX\beta + 2\lambda\beta = 0 & \\
(X^TX + 2\lambda I)\beta = X^Ty & \\
\beta = (X^TX + \lambda I)^{-1}X^Ty &
\end{aligned}$$

Therefore the estimator is $\hat{\beta}_{Ridge} = (X^TX + \lambda I)^{-1}X^Ty$.

b)

To find the expected value and the variance-covariance matrix of $\hat{\beta}_{Ridge}$ we need to remember the distribution of y, $y \sim N(X\beta, \sigma^2 I)$. Therefore we get the expected value:

$$E(\hat{\beta}_{Ridge}) = E((X^TX + \lambda I)^{-1}X^Ty) = (X^TX + \lambda I)^{-1}X^TE(y) = (X^TX + \lambda I)^{-1}X^TX\beta$$

and the variance-covariance matrix:

$$\begin{aligned}
Var(\hat{\beta}_{Ridge}) = Var((X^TX + \lambda I)^{-1}X^Ty) = & \quad \text{by property of the variance} \\
(X^TX + \lambda I)^{-1}X^TVar(y)((X^TX + \lambda I)^{-1}X^T)^T = & \quad \text{develop the expression} \\
\sigma^2(X^TX + \lambda I)^{-1}X^TX(X^TX + \lambda I)^{-1} &
\end{aligned}$$

c)

True, False, False (in a way yes?), True

d)

```
library(ISLR)
library(leaps)
library(glmnet)
```

We want to work with the *College* data. First we split it into a training and a testing set.

```
set.seed(1)

#make train and test set
train.ind = sample(1:nrow(College), 0.5 * nrow(College))
college.train = College[train.ind, ]
college.test = College[-train.ind, ]
str(College)
```

```
## 'data.frame':    777 obs. of  18 variables:
##  $ Private    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Apps       : num  1660 2186 1428 417 193 ...
##  $ Accept     : num  1232 1924 1097 349 146 ...
##  $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
##  $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
##  $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
##  $ F.Undergrad: num  2885 2683 1036 510 249 ...
##  $ P.Undergrad: num  537 1227 99 63 869 ...
##  $ Outstate   : num  7440 12280 11250 12960 7560 ...
##  $ Room.Board : num  3300 6450 3750 5450 4120 ...
##  $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
##  $ Personal   : num  2200 1500 1165 875 1500 ...
##  $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
##  $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
##  $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
##  $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
##  $ Expend     : num  7041 10527 8735 19016 10922 ...
##  $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

Now we will apply forward selection, using *Outstate* as a response. We have 18 variables including the response so we will obtain a model including up to 17 variables.

```
nb_predictors<-17
forward<-regsubsets(Outstate~.,college.train,nvmax=17,method="forward")
sum<-summary(forward)
```

In Figure 1 we can look at the Rss and the adjusted R^2 in order to pick how many variables give the optimal result. Remember that we would if the change is not very significant we would rather pick the simplest model. It seems like 5 variables would be good here.

```
par(mfrow=c(1,2))
plot(sum$rss,xlab="Number of Variables",ylab="RSS",type="l")
plot(sum$adjr2,xlab="Number of Variables",ylab="Adjusted RSq",type="l")
```

Below are the variables chosen when we decide to have 5.

```
# The exercise says: Choose a model according to one of the criteria that you know so I donOt think we

nb_selected_pred<-5
variables<-names(coef(forward,id=nb_selected_pred))
```

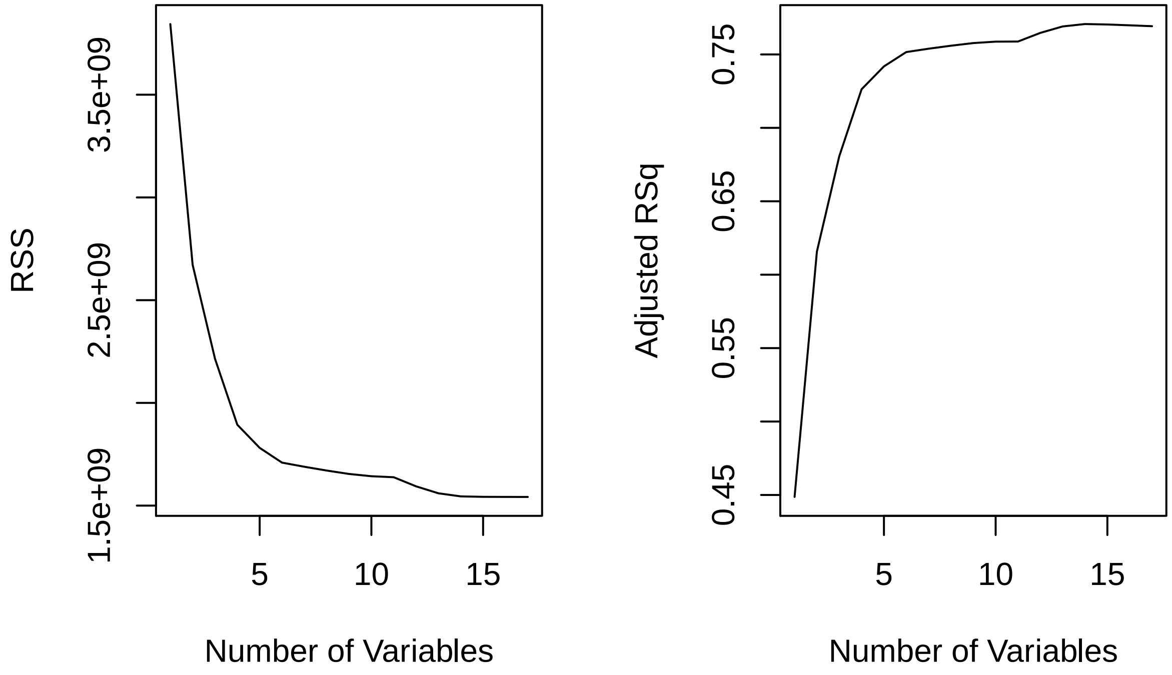


Figure 1: Comparison of models with different number of variables.

```
variables
```

```
## [1] "(Intercept)" "PrivateYes"  "Room.Board"  "perc.alumni" "Expend"
## [6] "Grad.Rate"
```

Now we can look at the summary of our final model as well as the MSE. The MSE is 4572478.

```
# can do this another way maybe?
model<-lm(Outstate~Private+Room.Board+Terminal+perc.alumni+Expend,college.train)
summary(model)
```

```
##
## Call:
## lm(formula = Outstate ~ Private + Room.Board + Terminal + perc.alumni +
##     Expend, data = college.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6445.5 -1420.8  -101.8  1409.1  9608.6
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4047.0191   712.4638  -5.680 2.67e-08 ***
## PrivateYes   2847.0626   302.6117   9.408  < 2e-16 ***
## Room.Board      1.2261     0.1226   9.998  < 2e-16 ***
## Terminal       43.0779     9.3515   4.607 5.59e-06 ***
## perc.alumni    75.1039    10.6011   7.085 6.76e-12 ***
## Expend          0.1983     0.0241   8.229 3.02e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2167 on 382 degrees of freedom
## Multiple R-squared:  0.7434, Adjusted R-squared:   0.74
## F-statistic: 221.3 on 5 and 382 DF,  p-value: < 2.2e-16
```

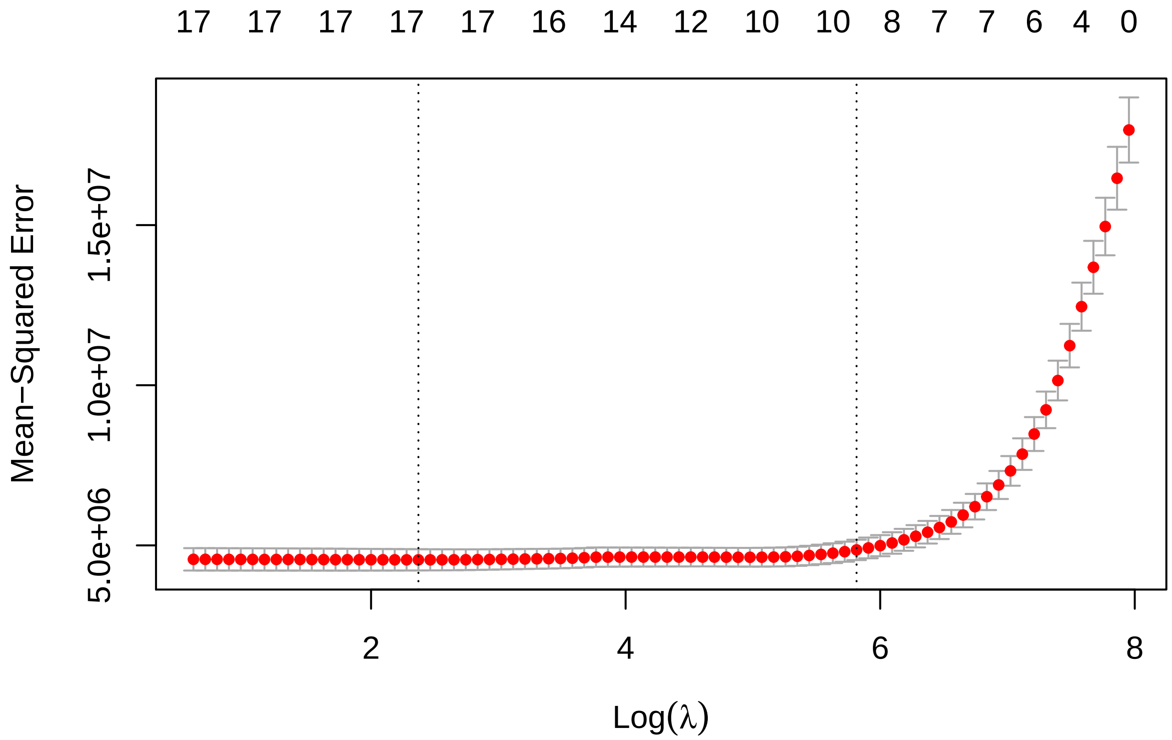
```
p<-predict(model,newdata=college.test)
error1<-mean(((college.test$Outstate)-p)^2)
error1
```

```
## [1] 4010675
```

e)

We will now select a model for the same dataset as in (d) but this time with the Lasso method.

```
x_train<-model.matrix(Outstate~.,college.train)[,-1]
y_train<-college.train$Outstate
x_test<-model.matrix(Outstate~.,college.test)[,-1]
y_test<-college.test$Outstate
mod.lasso = cv.glmnet(x_train,y_train,alpha=1) #alpha =1 is to use lasso
lambda.best = mod.lasso$lambda.min
plot(mod.lasso)
```



We used cross validation in order to find the best value for λ. Now we can predict and look at the MSE.

```
lasso_mod<-glmnet(x_train,y_train,alpha=1)
predictions<-predict(lasso_mod,s=lambda.best,newx=x_test)
error2<-mean((predictions-y_test)^2)
error2
```

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

The MSE on the test set is now 4264056, so lower than before. The variables that were not put to zero are displayed below.

```
c<-coef(lasso_mod,s=lambda.best,exact=TRUE)
inds<-which(c!=0)
variables<-row.names(c)[inds]
variables
```

```
##  [1] "(Intercept)" "PrivateYes"  "Apps"        "Accept"      "Enroll"
##  [6] "Top10perc"   "Top25perc"   "F.Undergrad" "P.Undergrad" "Room.Board"
## [11] "Books"       "Personal"    "PhD"         "Terminal"    "S.F.Ratio"
## [16] "perc.alumni" "Expend"      "Grad.Rate"
```

Problem 2

a)

FALSE, FALSE, TRUE, TRUE

b)

The basis functions for a cubic spline with knots at each quartile, of variable X are,

$$\begin{aligned} b_0(X) &= 1 & b_4(X) &= (X - q_1)^3_+ \\ b_1(X) &= x & b_5(x) &= (X - q_2)^3_+ \\ b_2(X) &= x^2 & b_6(X) &= (X - q_3)^3_+ \\ b_3(X) &= x^3 \end{aligned}$$

c)

We will now investigate the realtionship between Outstate and the 6 predictors as described in the problem.

//discuss which variables that seem to have a linear realtionship and benefit from non-linear transformations - splines

d)

Problem 3

a)

b)

c)

Problem 4

a)

b)

c)

d)

Problem 5

a)

b)

c)

d)

e)

f)

References

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. An Introduction to Statistical Learning with Applications in R. New York: Springer.