STAT 542: Homework 2

Spring 2020, by Yifan Shi (yifans16)

Due: Monday, Feb 17 by 11:59 PM

Contents

Question 1 [30 Points] Linear Model Selection

We will use the Boston Housing data for this question. The data is contained in the `mlbench` package. If you do not use R, you can download a `.csv` file from the course website. We will remove variables `medv`, `town` and `tract` from the data and use `cmedv` as the outcome. First, you need to standardize all variables marginally (only the covariates, not the outcome) to mean 0 and sample variation 1. Answer the following questions by performing linear regression and model selection.

a. [5 Points]

Perform a linear regression and obtain the ordinary least square estimators and their variances.

Answer:

```
library(mlbench)
library(tidyverse)
data("BostonHousing2")
df <- BostonHousing2 %>%
  select(-c(medv,town,tract))
df$chas <- as.numeric(df$chas)
df <- as.matrix(df)
xbar <- apply(df[,-3],2,mean)
Sx <- apply(df[,-3],2,sd)
ones <- matrix(1,506,1)
df[,-3] <- (df[,-3]-ones%*%xbar)/(ones%*%Sx)
```

The ols estimators are:

```
m1a <- lm(cmedv~.,data=as.data.frame(df))
m1a$coefficients
```

```
## (Intercept)         lon         lat        crim          zn       indus        chas
##  22.5288538  -0.2967352   0.2777157  -0.8992653   1.0860044   0.1045885   0.6548017
##         nox          rm         age         dis         rad         tax     ptratio
##  -1.8337152   2.6374222   0.0694618  -2.9478397   2.6706428  -2.1718248  -1.8989196
##           b       lstat
##   0.8377396  -3.8378999
```

The variances of ols estimators are:

```
(summary(m1a)$coefficients[,2])^2
```

```
## (Intercept)         lon         lat        crim          zn       indus        chas
##  0.04365969  0.06465208  0.05137158  0.07869648  0.10262204  0.17947492  0.04827557
##         nox          rm         age         dis         rad         tax     ptratio
##  0.21540709  0.08566990  0.14123974  0.19337482  0.33611403  0.39447244  0.08704823
##           b       lstat
##  0.05909841  0.12961030
```

b. [5 Points]

Starting from the full model, use stepwise regression with backward and BIC criterion to select the best model. Which variables are removed from the full model?

Answer:

```
m1b <- step(m1a,direction = "backward",trace=0,k=log(nrow(df)))
summary(m1b)
```

```
##
## Call:
## lm(formula = cmedv ~ crim + zn + chas + nox + rm + dis + rad +
##     tax + ptratio + b + lstat, data = as.data.frame(df))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.566  -2.686  -0.552   1.790  26.167
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.5289     0.2087 107.964  < 2e-16 ***
## crim         -0.9174     0.2794  -3.283 0.001099 **
## zn            1.0985     0.3126   3.514 0.000481 ***
## chas          0.6927     0.2150   3.221 0.001360 **
## nox          -2.0066     0.4060  -4.943 1.06e-06 ***
## rm            2.6550     0.2829   9.384  < 2e-16 ***
## dis          -3.2012     0.3876  -8.259 1.35e-15 ***
## rad           2.5822     0.5471   4.720 3.08e-06 ***
## tax          -2.0354     0.5633  -3.613 0.000333 ***
## ptratio      -1.9853     0.2769  -7.169 2.77e-12 ***
## b             0.8401     0.2419   3.473 0.000561 ***
## lstat        -3.7736     0.3356 -11.243  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.694 on 494 degrees of freedom
## Multiple R-squared:  0.7444, Adjusted R-squared:  0.7387
## F-statistic: 130.8 on 11 and 494 DF,  p-value: < 2.2e-16
```

`lon`, `lat`, `indus` and `age` are removed.

c. [5 Points]

Starting from this full model, use the best subset selection and list the best model of each model size.

Answer:

```
library(leaps)
library(knitr)
RSSleaps=regsubsets(df[,-3],df[,3], nvmax=15)
sumleaps <- summary(RSSleaps,matrix=T)
sumleaps.df <- as.data.frame(sumleaps$which)
sumleaps.df
```

```
##    (Intercept)   lon   lat  crim    zn indus  chas   nox    rm   age   dis   rad   tax
## 1         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## 3         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## 4         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 5         TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE
## 6         TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE
## 7         TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE
## 8         TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE
## 9         TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 10        TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 11        TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 12        TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 13        TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 14        TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 15        TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##    ptratio     b lstat
## 1    FALSE FALSE  TRUE
## 2    FALSE FALSE  TRUE
## 3     TRUE FALSE  TRUE
## 4     TRUE FALSE  TRUE
## 5     TRUE FALSE  TRUE
## 6     TRUE FALSE  TRUE
## 7     TRUE  TRUE  TRUE
## 8     TRUE  TRUE  TRUE
## 9     TRUE  TRUE  TRUE
## 10    TRUE  TRUE  TRUE
## 11    TRUE  TRUE  TRUE
## 12    TRUE  TRUE  TRUE
## 13    TRUE  TRUE  TRUE
## 14    TRUE  TRUE  TRUE
## 15    TRUE  TRUE  TRUE
```

d. [5 Points]

Use the BIC criterion to select the best model from part c). Which variables are removed from the full model?

Answer:

```
sumleaps$which[which.min(sumleaps$bic),]
```

```
## (Intercept)         lon         lat        crim          zn       indus        chas
##        TRUE       FALSE       FALSE        TRUE        TRUE       FALSE        TRUE
##         nox          rm         age         dis         rad         tax     ptratio
##        TRUE        TRUE       FALSE        TRUE        TRUE        TRUE        TRUE
##           b       lstat
##        TRUE        TRUE
```

`lon`, `lat`, `indus` and `age` are removed.

e. [10 Points]

Our solution is obtained based on the scaled X. Can you recover the original OLS estimates based on the original data? For this question, you can use information from the original design matrix. However, you cannot refit the linear regression. Provide a rigorous mathematical derivation and also validate that by comparing it to the `lm()` function on the original data.

Answer:

Since the response variable Y is not standardized, the estimated response $\hat{Y}$ should remain the same in both standardized and unstandardized linear models.

$$\hat{Y} = \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_i \frac{x_i - \bar{x_i}}{Sx_i}$$

where $\bar{x_i}$ and Sx_i are the column mean and standard deviation of the predictors.

After rearranging, the equation can be written as:

$$\hat{Y} = \hat{\beta}_0 - \sum_{i=1}^{k} \frac{\hat{\beta}_i \ \bar{x_i}}{Sx_i} + \sum_{i=1}^{k} \frac{\hat{\beta}_i}{Sx_i} x_i$$

while

$$\hat{\beta}_0 - \sum_{i=1}^{k} \frac{\hat{\beta}_i \ \bar{x_i}}{Sx_i}$$

is the intercept and

$$\frac{\hat{\beta}_i}{Sx_i}$$

are the coefficients in the unstandardized model.

```
df2 <- BostonHousing2 %>%
  select(-c(medv,town,tract))
df2$chas <- as.numeric(df2$chas)
m1e <- lm(cmedv~.,data=df2)

intercept <- m1a$coefficients[1] - sum(m1a$coefficients[2:16]*xbar/Sx)
coefficient <- m1a$coefficients[2:16]/Sx
recovered <- c(intercept,coefficient)
```

```
rm(intercept,coefficient)
original <- m1e$coefficients
kable(cbind(original,recovered))
```

	original	recovered
(Intercept)	-437.5738685	-437.5738685
lon	-3.9352016	-3.9352016
lat	4.4954414	4.4954414
crim	-0.1045469	-0.1045469
zn	0.0465648	0.0465648
indus	0.0152453	0.0152453
chas	2.5780197	2.5780197
nox	-15.8245766	-15.8245766
rm	3.7537117	3.7537117
age	0.0024677	0.0024677
dis	-1.3999266	-1.3999266
rad	0.3067145	0.3067145
tax	-0.0128863	-0.0128863
ptratio	-0.8771212	-0.8771212
b	0.0091762	0.0091762
lstat	-0.5374411	-0.5374411

Question 2 [70 Points] Ridge Regression and Scaling Issues

For this question, you can **ONLY** use the base package. We will use the dataset from Question 1 a). However, you should further standardize the outcome variable `cmedv` to mean 0 and sample variance 1. Hence, no intercept term is needed when you fit the model.

a. [30 points]

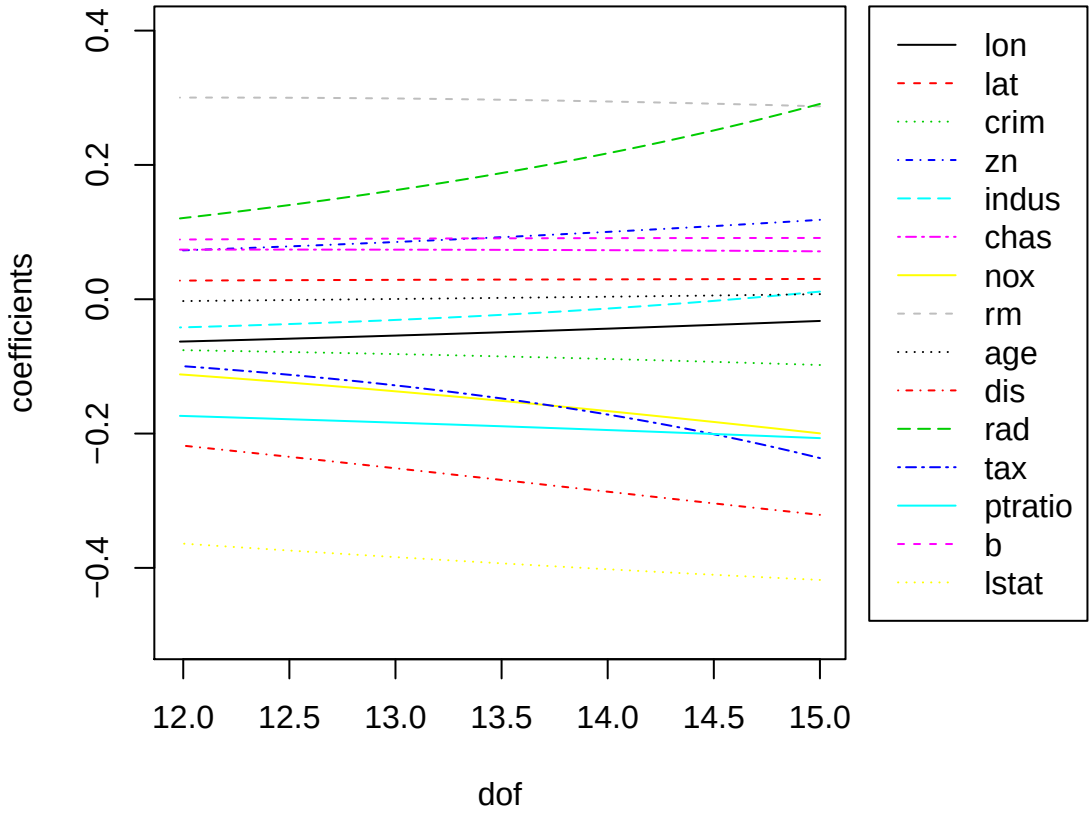
First, fit a ridge regression with your own code, with the objective function $||\mathbf{y}-\mathbf{X}\boldsymbol{\beta}||^2+\lambda||\boldsymbol{\beta}||^2$. * You should consider a grid of 100 penalty λ values. Your choice of lambda grid can be flexible. However, they should be appropriate for the scale of the objective function. * Calculate the degrees of freedom and the leave-one-out cross-validation (computationally efficient version) based on your choice of the penalty. * Report details of your model fitting result. In particular, you should produce a plot similar to page 25 in the lecture note `Penalized`

Answer:

```
X <- df[,-3]
Y <- df[,3]
Y <- (Y-mean(Y))/sd(Y)
lambda <- seq(0,49.5,by=0.5)
coef <- matrix(NA,length(lambda),ncol(X))
dof <- rep(NA,length(lambda))
CV <- rep(NA,length(lambda))
for (i in 1:length(lambda)){
  div <- solve(t(X)%*%X+diag(lambda[i],ncol(X),ncol(X)))
  coef[i,] = div%*%t(X)%*%Y
```

```
  dof[i] = sum(diag(X%*%div%*%t(X)))
  CV[i] = sum(((Y-X%*%coef[i,])/(1-diag(X%*%div%*%t(X))))^2)
}
```

```
par(mar=c(5,5,3,7),xpd=T)
plot(dof,coef[,1],type="l",col=1,lty=1,ylim=range(-0.5,0.4),ylab="coefficients")
for (j in 2:15){
  lines(dof,coef[,j],type="l",col=j,lty=j)
}
legend("topright",inset=c(-.35,0),c(colnames(X)),col=c(1:15),lty=c(1:15))
```



The λ gives the smallest LOOCV is:

```
lambda[which.min(CV)]
```

```
## [1] 5.5
```

b. [25 points]

Following the setting of part a), with $\lambda = 10$, recover the original solution based on the unstandardized data (with intercept). Again, you cannot refit the model on the original data. Provide a rigorous mathematical derivation. In this case, is your solution the same as a ridge model (with your previous code) fitted on the original data? Make sure that you check the model fitting results from either model. What is the difference? Please list all possible reasons that may cause this difference.

Answer:

According to the relation that

$$\hat{Y}_{stan} = \frac{\hat{Y}_{unstan} - \bar{Y}}{Sy}$$

and

$$\hat{Y}_{stan} = \sum_{i=1}^{k} \hat{\beta}_i \frac{x_i - \bar{x_i}}{Sx_i}$$

we can get

$$\hat{Y}_{unstan} = \bar{Y} + \sum_{i=1}^{k} Sy\hat{\beta}_i \frac{x_i - \bar{x_i}}{Sx_i}$$

where $\bar{Y}$ and Sy are the mean and standard deviation of the unstandardized response.

After rearranging, the equation can be written as:

$$\hat{Y}_{unstan} = \bar{Y} - \sum_{i=1}^{k} \frac{Sy}{Sx_i}\hat{\beta}_i\bar{x_i} + \sum_{i=1}^{k} \frac{Sy}{Sx_i}\hat{\beta}_i x_i$$

while

$$\bar{Y} - \sum_{i=1}^{k} \frac{Sy}{Sx_i}\hat{\beta}_i\bar{x_i}$$

is the intercept and

$$\frac{Sy}{Sx_i}\hat{\beta}_i$$

are the coefficients in the recovered model.

```
#undtandardized y_hat from unstandardized x
X_u <- as.matrix(df2[,-3])
div_u <- solve(t(X_u)%*%X_u+diag(10,ncol(X_u),ncol(X_u)))
yhat_u1 <- as.vector(X_u%*%div_u%*%t(X_u)%*%df[,3])

#undtandardized y_hat from standardized beta_hat
div_s <- solve(t(X)%*%X+diag(10,ncol(X),ncol(X)))
coef_s <- div_s%*%t(X)%*%Y
ybar <- matrix(mean(df[,3]),nrow(df),1)
beta1_u <- coef_s/Sx*sd(df[,3])
beta0_u <- ybar - ones%*%(xbar/Sx)%*%coef_s * sd(df[,3])
yhat_u2 <- as.vector(beta0_u + X_u%*%beta1_u)
difference <- yhat_u1-yhat_u2
rm(X_u,div_u)
kable(head(cbind(yhat_u1,yhat_u2,difference)))
```

yhat_u1	yhat_u2	difference
30.59789	29.74208	0.8558140
24.60654	24.83053	-0.2239898
30.44401	30.26505	0.1789544
29.25690	28.62443	0.6324638
28.37982	27.97328	0.4065481
25.74567	25.26789	0.4777759

The reason that causes this difference might be the shrinkage of intercept.

c. [15 points]

A researcher is interested in only penalizing a subset of the variables, and leave the rest of them unpenalized. In particular, the categorical variables `zn`, `chas`, `rad` should not be penalized. You should use the data in part 2), which does not concern the intercept. Following the derivation during our lecture: * Write down the objective function of this new model * Derive the theoretical solution of this model and implement it with $\lambda = 10$ * Check your model fitting results and report the residual sum of squares

Answer:

The objective function is: $F = (Y - X_1\beta_1 - X_2\beta_2)^T(Y - X_1\beta_1 - X_2\beta_2) + \lambda\beta_1^T\beta_1$

In order to find the minimum, we need to take partial derivatives based on β_1 and β_2 :

$$\begin{aligned}\frac{\partial F}{\partial \beta_1} &= -X_1^T(Y - X_1\beta_1 - X_2\beta_2) + \lambda\beta_1 \\ &= -X_1^TY + X_1^TX_1\beta_1 + X_1^TX_2\beta_2 + \lambda\beta_1\end{aligned}$$

$$\begin{aligned}\frac{\partial F}{\partial \beta_2} &= -X_2^T(Y - X_1\beta_1 - X_2\beta_2) \\ &= -X_2^TY + X_2^TX_1\beta_1 + X_2^TX_2\beta_2\end{aligned}$$

Then set these partial derivatives to be zero, we will get the estimated parameters:

$$\hat{\beta_2} = (X_2^TX_2)^{-1}(X_2^TY - X_2^TX_1\hat{\beta_1})$$

$$\begin{aligned}\hat{\beta_1} &= (X_1^TX_1 - \lambda I)^{-1}(X_1^TY - X_1^TX_2\hat{\beta_2}) \\ &= (X_1^TX_1 - \lambda I)^{-1}(X_1^TY - X_1^TX_2(X_2^TX_2)^{-1}(X_2^TY - X_2^TX_1\hat{\beta_1}) \\ &= (X_1^TX_1 - \lambda I)^{-1}X_1^TY - (X_1^TX_1 - \lambda I)^{-1}X_1^TX_2(X_2^TX_2)^{-1}X_2^TY \\ &\quad + (X_1^TX_1 - \lambda I)^{-1}X_1^TX_2(X_2^TX_2)^{-1}X_2^TX_1\hat{\beta_1} \\ &= [I - (X_1^TX_1 - \lambda I)^{-1}X_1^TX_2(X_2^TX_2)^{-1}X_2^TX_1]^{-1} \\ &\quad * [(X_1^TX_1 - \lambda I)^{-1}X_1^TY - (X_1^TX_1 - \lambda I)^{-1}X_1^TX_2(X_2^TX_2)^{-1}X_2^TY]\end{aligned}$$

The model fitting results:

```
x1 <- X[,-c(4,6,11)]
x2 <- X[,c(4,6,11)]
p1 <- ncol(x1)
p2 <- ncol(x2)
beta1hat <- solve(diag(p1)-solve(t(x1)%*%x1-diag(10,p1,p1))%*%
                    t(x1)%*%x2%*%solve(t(x2)%*%x2)%*%t(x2)%*%
                    x1)%*%(solve(t(x1)%*%x1-diag(10,p1,p1))%*%
                    t(x1)%*%Y-solve(t(x1)%*%x1-diag(10,p1,p1))%*%
                    t(x1)%*%x2%*%solve(t(x2)%*%x2)%*%t(x2)%*%Y)
beta2hat <- solve(t(x2)%*%x2)%*%(t(x2)%*%Y-t(x2)%*%x1%*%beta1hat)
head(x1%*%beta1hat+x2%*%beta2hat)
```

```
##         [,1]
## 1 0.7808897
## 2 0.2791694
## 3 0.8783161
## 4 0.6617611
## 5 0.5822771
## 6 0.3024907
```

The RSS of this model is:

```
sum((Y-x1%*%beta1hat-x2%*%beta2hat)^2)
```

```
## [1] 128.7978
```