Homework 2: Solutions

E6690: Statistical Learning for Bio & Info Systems

P1. It is well known that ridge regression tends to give similar coefficient values to correlated variables, whereas the lasso may give quite different coefficient values to correlated variables. We will now explore this property in a very simple setting.

Suppose that $n = 2$, $p = 2$, $x_{11} = x_{12}$, $x_{21} = x_{22}$. Furthermore, suppose that $y_1 + y_2 = 0$ and $x_{11} + x_{21} = 0$ and $x_{12} + x_{22} = 0$, so that the estimate for the intercept in a least squares, ridge regression, or lasso model is zero: $\hat{\beta}_0 = 0$.

(a) (2pt) Write out the ridge regression optimization problem in this setting.

Answer: The general form of ridge regression optimization looks like

$$\min \quad \sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_j \right)^2 + \lambda \sum_{i=1}^{p} \hat{\beta}_i^2,$$

with $\hat{\beta}_0 = 0$ and $n = p = 2$, it becomes

$$\min \quad (y_1 - \hat{\beta}_1 x_{11} - \hat{\beta}_2 x_{12})^2 + (y_2 - \hat{\beta}_1 x_{21} - \hat{\beta}_2 x_{22})^2 + \lambda(\hat{\beta}_1^2 + \hat{\beta}_2^2).$$

(b) (2pt) Argue that in this setting, the ridge coefficient estimates satisfy $\hat{\beta}_1 = \hat{\beta}_2$.

Answer: We are given that $x_{11} = x_{12} = x_1$ and $x_{21} = x_{22} = x_2$. Take deviations of the above objective function with respect to $\hat{\beta}_1$ and $\hat{\beta}_2$ and set them equal to zero. That yields

$$\hat{\beta}_1^* = \frac{x_1 y_1 + x_2 y_2 - \hat{\beta}_2^*(x_1^2 + x_2^2)}{\lambda + x_1^2 + x_2^2},$$
$$\hat{\beta}_2^* = \frac{x_1 y_1 + x_2 y_2 - \hat{\beta}_1^*(x_1^2 + x_2^2)}{\lambda + x_1^2 + x_2^2}.$$

Symmetry in these two expressions suggests that $\hat{\beta}_1^* = \hat{\beta}_2^*$.

(c) (2pt) Write out the lasso optimization problem in this setting.

Answer:

$$\min \quad (y_1 - \hat{\beta}_1 x_{11} - \hat{\beta}_2 x_{12})^2 + (y_2 - \hat{\beta}_1 x_{21} - \hat{\beta}_2 x_{22})^2 + \lambda(|\hat{\beta}_1| + |\hat{\beta}_2|).$$

(d) (4pt) Argue that in this setting, the lasso coefficients $\hat{\beta}_1$ and $\hat{\beta}_2$ are not unique-in other words, there are many possible solutions to the optimization problem in (c). Describe these solutions.

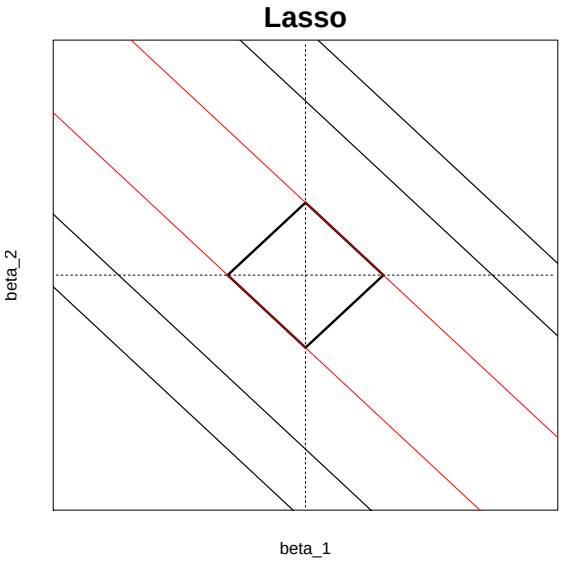
Answer: The lasso constraint takes the form $|\hat{\beta}_1| + |\hat{\beta}_2| \leq s$, which when plotted take the familiar shape of a diamond centered at $(0, 0)$. By assuming $x_{11} = x_{12}$, $x_{21} = x_{22}$, $x_{11} + x_{21} = 0$, $x_{12} + x_{22} = 0$ and $y_1 + y_2 = 0$, the constrained objective function reduces to

$$\min \quad 2(y_1 - (\hat{\beta}_1 + \hat{\beta}_2)x_{11})^2.$$

This constrained optimization has a solution: $\hat{\beta}_1 + \hat{\beta}_2 = y_1/x_{11}$. This is a line parallel to the edge of lasso-diamond $\hat{\beta}_1 + \hat{\beta}_2 = s$. Now solutions to the original problem are contours of the function

$(y_1 - (\hat{\beta}_1 + \hat{\beta}_2)x_{11})^2$ that touch the $\hat{\beta}_1 + \hat{\beta}_2 = s$. As a result, the entire edge $\hat{\beta}_1 + \hat{\beta}_2 = s$ is a potential solution to the problem. Similar argument can be made for opposite edge: $\hat{\beta}_1 + \hat{\beta}_2 = -s$. Therefore, this problem does not have a unique solution. The general form of solution is given by two line segments:

$$\begin{aligned} \hat{\beta}_1 + \hat{\beta}_2 &= s, \qquad \hat{\beta}_1 \geq 0, \hat{\beta}_2 \geq 0, \\ \hat{\beta}_1 + \hat{\beta}_2 &= -s, \qquad \hat{\beta}_1 \leq 0, \hat{\beta}_2 \leq 0. \end{aligned}$$



P2. Assume that we are performing regression without an intercept. In this case with $n = p$, the usual least squares problem simplifies to finding $\beta_1, \ldots, \beta_p$ that minimize

$$\sum_{j=1}^{p} (y_i - \beta_j)^2.$$

The least squares solution is given by $\hat{\beta}_j = y_j$.
And in this setting, ridge regression amounts to finding $\beta_1, \ldots, \beta_p$ such that

$$\sum_{j=1}^{p} (y_i - \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{1}$$

is minimized, and the lasso amounts to finding the coefficients such that

$$\sum_{j=1}^{p} (y_i - \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2}$$

is minimized. One can show that in this setting, the ridge regression estimates take the form

$$\hat{\beta}_j^R = y_j/(1 + \lambda), \tag{3}$$

and the lasso estimates take the form

$$\hat{\beta}_j^L = \begin{cases} y_j - \lambda/2 & \text{if } y_j > \lambda/2; \\ y_j + \lambda/2 & \text{if } y_j < -\lambda/2; \\ 0 & \text{if } |y_j| \leq \lambda/2. \end{cases} \tag{4}$$

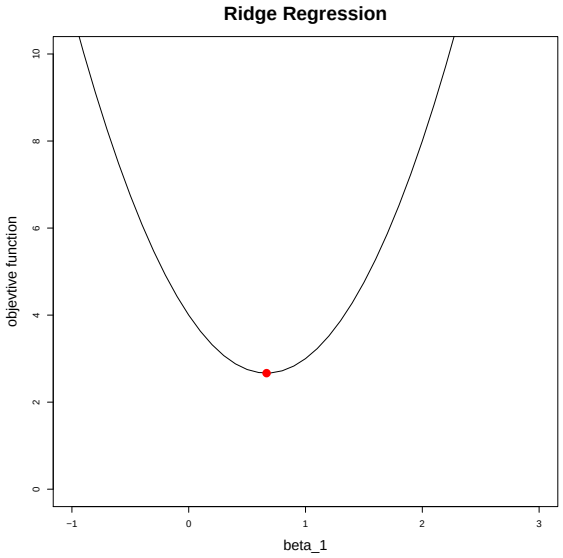
(a) (5pt) Consider (1) with $p = 1$. For some choice of y_1 and $\lambda > 0$, plot (1) as a function of β_1. Your plot should confirm that (1) is solved by (3).

Answer: For $p = 1$, (1) looks like $(y - \beta_1)^2 + \lambda\beta_1^2 = (1 + \lambda)\beta_1^2 - 2y\beta_1 + y^2$. This is a parabola with the minimum achieved at $y/(1 + \lambda)$. Indeed, this value solves

$$0 = ((1 + \lambda)\beta_1^2 - 2y\beta_1 + y^2)' = (1 + \lambda)\beta_1 - 2y_1.$$

Now, assume $y = 2$ and $\lambda = 2$.

```
> y<-2
> lambda<- 2
> beta_1<-seq(-10,10,0.1)
> func<-(y-beta_1)^2+lambda*beta_1 ^2
> sol<-y/(1+lambda)
> sol_func<-(y-sol)^2+lambda*sol ^2
> pdf(file="2-a.pdf",width = 10,height = 10)
> plot(beta_1,func,type="l",xlab = "beta_1",ylab = "objevtive function",xlim=c(-1,3)
    ,ylim=c(0,10),cex.lab=1.5)
> title("Ridge Regression",cex.main=2)
> points(sol,sol_func,col="red",lwd=7)
> dev.off()
```



The red point shows that function is indeed minimized at point $\beta_1 = y/(1 + \lambda) = \frac{2}{3}$.

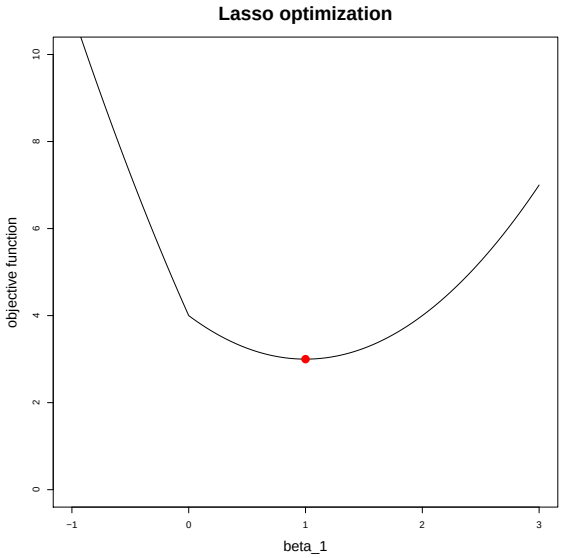
(b) (5pt) Consider (2) with $p = 1$. For some choice of y_1 and $\lambda > 0$, plot (2) as a function of β_1. Your plot should confirm that (2) is solved by (4).

Answer: For $p = 1$, (2) takes the form

$$\begin{cases} (y - \beta_1)^2 + \lambda\beta_1 & \beta_1 \le 0, \\ (y - \beta_1)^2 - \lambda\beta_1 & \beta_1 < 0. \end{cases}$$

We plot the function with $y = 2$ and $\lambda = 2$.

```
> y<-2
> lambda<- 2
> beta_1<-seq(-3,3,0.01)
> func<-(y-beta_1)^2+lambda*abs(beta_1)
> sol<-y-lambda/2
> sol_func<-(y-sol)^2+lambda*abs(sol)
> pdf(file="2-b.pdf",width = 10,height = 10)
> plot(beta_1,func,type="l",xlab = "beta_1",ylab = "objective function",xlim=c(-1,3)
    ,ylim=c(0,10),cex.lab=1.5)
> title("Lasso optimization",cex.main=2)
> points(sol,sol_func,col="red",lwd=7)
> dev.off()
```



The red point in the plot shows that function is minimized at $\beta_1 = y - \lambda/2 = 1$.

P3. In this problem, we will derive the Bayesian connection to the lasso and ridge regression.

(a) (2pt) Suppose that $y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i$ where $\epsilon_1, \ldots, \epsilon_n$ are independent and identically distributed from a $\mathcal{N}(0, \sigma^2)$ distribution. Write out the likelihood for the data.

Answer: The likelihood function is

$$L(\theta|\boldsymbol{\beta}) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y_i - (\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}))^2}{2\sigma^2}\right]$$
$$= \frac{1}{(\sigma\sqrt{2\pi})^n} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right)\right)^2\right].$$

(b) (3pt) Assume the following prior for $\boldsymbol{\beta}$: $\beta_1, \ldots, \beta_p$ are independent and identically distributed according to a double-exponential distribution with mean 0 and common scale parameter b; i.e, $p(\boldsymbol{\beta}) = \frac{1}{2b}\exp(-|\boldsymbol{\beta}|/b)$ and $|\boldsymbol{\beta}| = \sum_{j=1}^{p} |\beta_j|$. Write out the posterior for $\boldsymbol{\beta}$ in this setting.

Answer: The posterior with double-exponential distribution with mean 0 and common scale parameter b is:

$$f(\boldsymbol{\beta}|x,y) \propto f(y|x,\boldsymbol{\beta})p(\boldsymbol{\beta}|x) = f(y|x,\boldsymbol{\beta})p(\boldsymbol{\beta}).$$

Substituting likelihood function from (a) and this density function gives us

$$\begin{aligned} f(y|x,\boldsymbol{\beta})p(\boldsymbol{\beta}) &= \frac{1}{(\sigma\sqrt{2\pi})^n}\ \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right)\right)^2\right]\frac{1}{2b}\exp(-|\boldsymbol{\beta}|/b) \\ &= \frac{1}{(\sigma\sqrt{2\pi})^n}\frac{1}{2b}\exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right)\right)^2 - \frac{|\boldsymbol{\beta}|}{b}\right]. \end{aligned}$$

(c) (5pt) Argue that the lasso estimate is the mode for β under this posterior distribution.

Answer: Showing this is the same thing as showing that the most likely value for $\boldsymbol{\beta}$ is given by the lasso solution with a certain λ. Let's start by simplifying the likelihood function by taking the logarithm of both sides:

$$\log f(y|y,\boldsymbol{\beta})p(\boldsymbol{\beta}) = \log\left(\frac{1}{2b(\sigma\sqrt{2\pi})^n}\right) - \left(\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left[y_i - (\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij})\right]^2 + \frac{|\boldsymbol{\beta}|}{b}\right).$$

We want to maximize the posterior, which means

$$\begin{aligned} \text{argmax}_{\boldsymbol{\beta}} f(\boldsymbol{\beta}|x,y) &= \text{argmax}_{\boldsymbol{\beta}}\ \log\left(\frac{1}{2b(\sigma\sqrt{2\pi})^n}\right) - \left[\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - (\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij})\right)^2 + \frac{|\boldsymbol{\beta}|}{b}\right] \\ &= \text{argmin}_{\boldsymbol{\beta}}\ \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - (\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij})\right)^2 + \frac{1}{b}\sum_{j=1}^{p}|\beta_j| \\ &= \text{argmin}_{\boldsymbol{\beta}}\ \sum_{i=1}^{n}\left(y_i - (\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij})\right)^2 + \frac{2\sigma^2}{b}\sum_{j=1}^{p}|\beta_j|. \end{aligned}$$

By letting $\lambda = 2\sigma^2/b$, we can see that we end up with

$$\begin{aligned} \text{argmax}_{\boldsymbol{\beta}} f(\boldsymbol{\beta}|x,y) &= \text{argmin}_{\boldsymbol{\beta}}\ \sum_{i=1}^{n}\left(y_i - (\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij})\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \\ &= \text{argmin}_{\boldsymbol{\beta}}\ \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|, \end{aligned}$$

which is the lasso equation. Thus, we know that when the posterior comes from a double-exponential distribution with mean 0 and common scale parameter b, the mode for β is given by the lasso solution when $\lambda = 2\sigma^2/b$.

(d) (5pt) Now assume that the following prior for $\beta : \beta_1, \ldots, \beta_p$ are independent and identically distributed according to a normal distribution with mean 0 and variance c. write out the posterior for β in this setting.

Answer: The posterior distribution in this case is

$$f(\boldsymbol{\beta}|x,y) \propto f(y|x,\boldsymbol{\beta})p(\boldsymbol{\beta}|x) = f(y|x,\boldsymbol{\beta})p(\boldsymbol{\beta}).$$

The probability distribution function then becomes

$$p(\boldsymbol{\beta}) = \prod_{i=1}^{p} p(\beta_i) = \prod_{i=1}^{p} \frac{1}{\sqrt{2c\pi}} \exp\left(-\frac{\beta_i^2}{2c}\right) = \left(\frac{1}{\sqrt{2c\pi}}\right)^p \exp\left(-\frac{1}{2c}\sum_{i=1}^{p}\beta_i^2\right).$$

Substituting the value from (a) and this density function, we have

$$\begin{aligned} f(y|x,\boldsymbol{\beta})p(\boldsymbol{\beta}) &= \frac{1}{(\sigma\sqrt{2\pi})^n} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right)\right)^2\right]\left(\frac{1}{\sqrt{2c\pi}}\right)^p \exp\left(-\frac{1}{2c}\sum_{i=1}^{p}\beta_i^2\right) \\ &= \frac{1}{(\sigma\sqrt{2\pi})^n}\frac{1}{(\sqrt{2c\pi})^p} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right)\right)^2 - \frac{1}{2c}\sum_{i=1}^{p}\beta_i^2\right]. \end{aligned}$$

(e) (5pt) Argue that the ridge regression estimate is both the mode and the mean for $\boldsymbol{\beta}$ under this posterior distribution.

Answer: Showing this is the same thing as showing that the most likely value for $\boldsymbol{\beta}$ is given by the lasso solution with a certain λ. Let's start by simplifying it by taking the logarithm of both sides

$$\log f(y|x,\boldsymbol{\beta})p(\boldsymbol{\beta}) = \log\left(\frac{1}{(\sigma\sqrt{2\pi})^n}\frac{1}{(\sqrt{2c\pi})^p}\right) - \left[\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right)\right)^2 + \frac{1}{2c}\sum_{i=1}^{p}\beta_i^2\right].$$

We want to maximize the posterior, which means

$$\begin{aligned} \text{argmax}_{\boldsymbol{\beta}} f(\boldsymbol{\beta}|x,y) &= \text{argmax}_{\boldsymbol{\beta}}\ \log\left(\frac{1}{(\sigma\sqrt{2\pi})^n}\frac{1}{(\sqrt{2c\pi})^p}\right) - \left[\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right)\right)^2 + \frac{1}{2c}\sum_{i=1}^{p}\beta_i^2\right] \\ &= \text{argmin}_{\boldsymbol{\beta}}\ \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right)\right)^2 + \frac{1}{2c}\sum_{i=1}^{p}\beta_i^2. \end{aligned}$$

By letting $\lambda = \sigma^2/c$, we end up with

$$\begin{aligned} \text{argmax}_{\boldsymbol{\beta}} f(\boldsymbol{\beta}|x,y) &= \text{argmin}_{\boldsymbol{\beta}}\ \left(\frac{1}{2\sigma^2}\right)\left[\sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right)\right)^2 + \lambda\sum_{i=1}^{p}\beta_i^2\right] \\ &= \text{argmin}_{\boldsymbol{\beta}}\ \text{RSS} + \lambda\sum_{i=1}^{p}\beta_i^2. \end{aligned}$$

which is the ridge equation. Thus, we know that when the posterior comes from a normal distribution with mean 0 and variance c, the mode for $\boldsymbol{\beta}$ is given by the ridge regression solution when $\lambda = \sigma^2/c$.

P4. In this exercise, we will generate simulated data, and will then use this data to perform best subset selection.

(a) (2pt) Use the `rnorm()` function to generate a predictor X of length $n = 100$, as well as a noise vector ϵ of length $n = 100$.

Answer:

```
> set.seed(1)
> x<-rnorm(100)
> eps<-rnorm(100)
```

(b) (3pt) Generate a response vector Y of length $n = 100$ according to the model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon,$$

where β_0, β_1, β_2, and β_3 are constants of your choice.

Answer: We choose $\beta_0 = 3$, $\beta_1 = 2$, $\beta_2 = -3$ and $\beta_3 = 0.3$.

```
> beta_0<-3
> beta_1<-2
> beta_2<--3
> beta_3<-0.3
> y<-beta_0+beta_1*x+beta_2*x^2+beta_3*x^3+eps
```

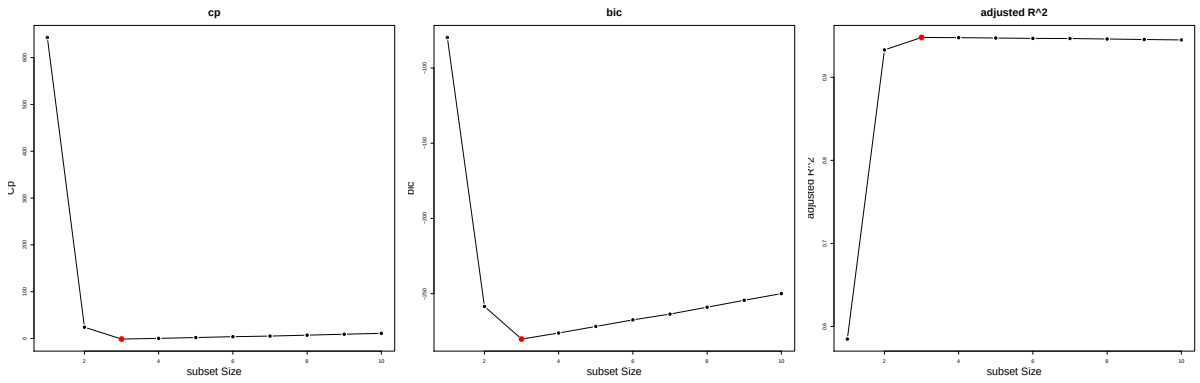
(c) (5pt) Use the `regsubsets()` function to perform best subset selection in order to choose the best model containing the predictors $X, X^2, \ldots, X^{10}$. What is the best model obtained according to C_p, BIC, and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained. Note you will need to use the `data.frame()` function to create a single data set containing both X and Y.

Answer:

```
> library(leaps)
> data1 <- data.frame(y = y, x = x)
> model1 <- regsubsets(y ~ poly(x, 10, raw = T), data = data1, nvmax = 10)
> model1_summary <- summary(model1)
> which.min(model1_summary$cp)
## [1] 3
> which.min(model1_summary$bic)
## [1] 3
> which.max(model1_summary$adjr2)
## [1] 3
> pdf(file="4-c.pdf",width = 25,height = 8)
> par(mfrow = c(1, 3))
> plot(model1_summary$cp, xlab = "subset Size",ylab = "Cp", type = "b"
    ,pch=19, cex.lab=2)
> title("cp",cex.main=2)
> points(3, model1_summary$cp[3], col = "red",lwd=7)
> plot(model1_summary$bic, xlab = "subset Size",ylab = "bic", type = "b"
   ,pch=19,cex.lab=2)
> title("bic",cex.main=2)
> points(3, model1_summary$bic[3], col = "red",lwd=7)
```

```
> plot(model1_summary$adjr2, xlab = "subset Size",ylab = "adjusted R^2",
     type = "b",pch=19,cex.lab=2)
> title("adjusted R^2",cex.main=2)
> points(3, model1_summary$adjr2[3], col = "red",lwd=7)
> dev.off()
> coefficients(model1, id = 3)
##          (Intercept) poly(x, 10, raw = T)1
##           3.07627412            2.35623596
##poly(x, 10, raw = T)2 poly(x, 10, raw = T)7
##          -3.16514887            0.01046843
```

All statistics pick X^7 over X^3. Note that $|\beta_0 - \hat{\beta}_0| = 0.7627$, $|\beta_1 - \hat{\beta}_1| = 0.3562$ and $|\beta_2 - \hat{\beta}_2| = 0.1651$.

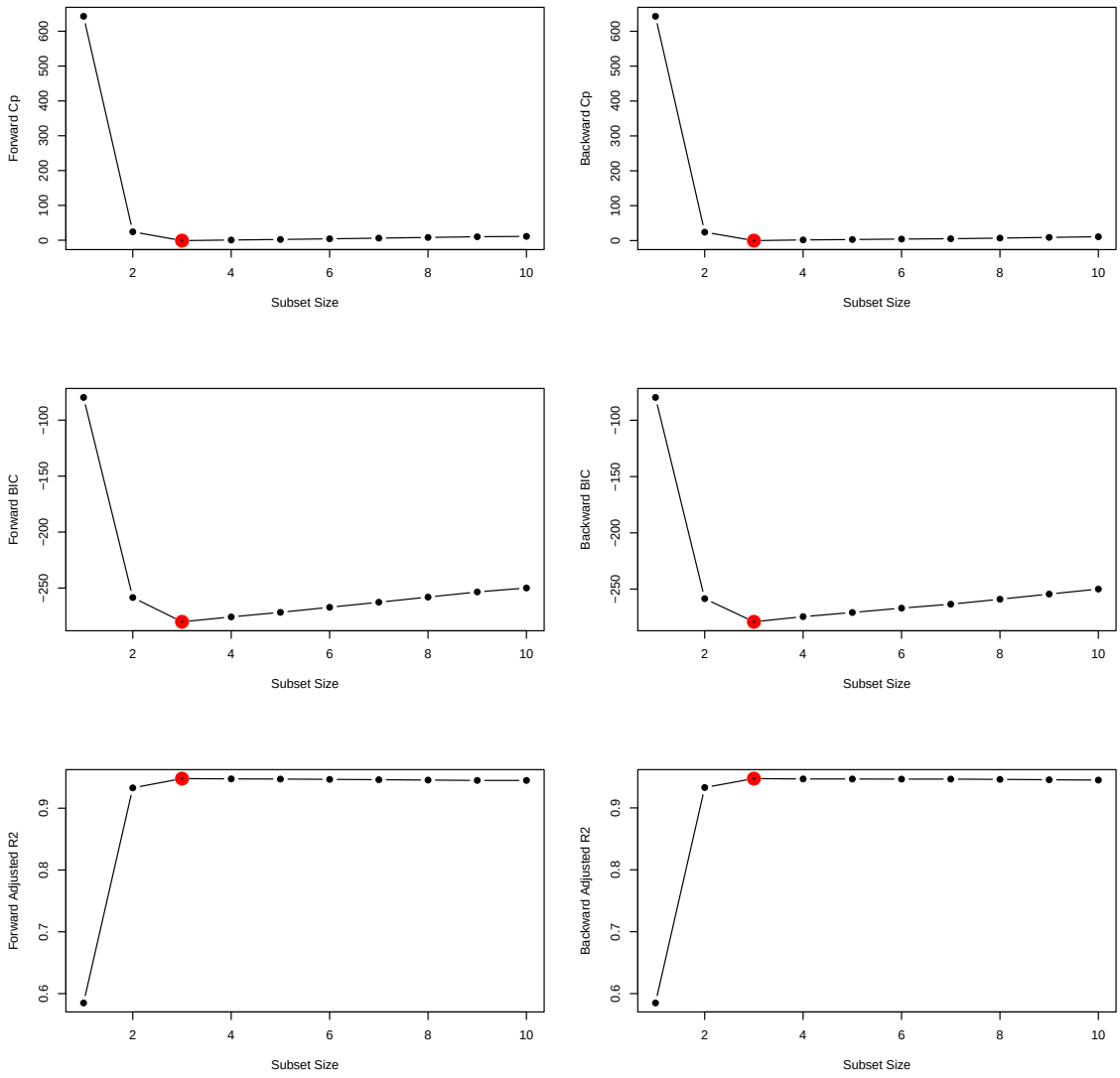


(d) (5pt) Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the result in (c)?

Answer:

```
> model1_fwd <- regsubsets(y ~ poly(x, 10, raw = T), data = data1, nvmax = 10,
+                      method = "forward")
> model1_bwd <- regsubsets(y ~ poly(x, 10, raw = T), data = data1, nvmax = 10,
+                      method = "backward")
> fwd.summary <- summary(model1_fwd)
> bwd.summary <- summary(model1_bwd)
> which.min(fwd.summary$cp)
## [1] 3
> which.min(bwd.summary$cp)
## [1] 3
> which.min(fwd.summary$bic)
## [1] 3
> which.min(bwd.summary$bic)
## [1] 3
> which.max(fwd.summary$adjr2)
## [1] 3
> which.max(bwd.summary$adjr2)
## [1] 3
> pdf(file="4-d.pdf",width = 10,height = 10)
```

```
> par(mfrow = c(3, 2))
> plot(fwd.summary$cp, xlab = "Subset Size", ylab = "Forward Cp",
    pch=19,type = "b")
> points(3, fwd.summary$cp[3], col = "red", lwd = 7)
> plot(bwd.summary$cp, xlab = "Subset Size", ylab = "Backward Cp",
    pch=19, type = "b")
> points(3, bwd.summary$cp[3], col = "red", lwd = 7)
> plot(fwd.summary$bic, xlab = "Subset Size", ylab = "Forward BIC",
+      type = "b",pch=19)
> points(3, fwd.summary$bic[3], col = "red", lwd = 7)
> plot(bwd.summary$bic, xlab = "Subset Size", ylab = "Backward BIC",
+      type = "b",pch=19)
> points(3, bwd.summary$bic[3], col = "red", lwd = 7)
> plot(fwd.summary$adjr2, xlab = "Subset Size", ylab = "Forward Adjusted R2",
+      type = "b",pch=19)
> points(3, fwd.summary$adjr2[3], col = "red", lwd = 7)
> plot(bwd.summary$adjr2, xlab = "Subset Size", ylab = "Backward Adjusted R2",
+       type = "b",pch=19)
> points(3, bwd.summary$adjr2[3], col = "red", lwd = 7)
> dev.off()
```



We see that all statistics pick 3 variable models. Here are the coefficients:

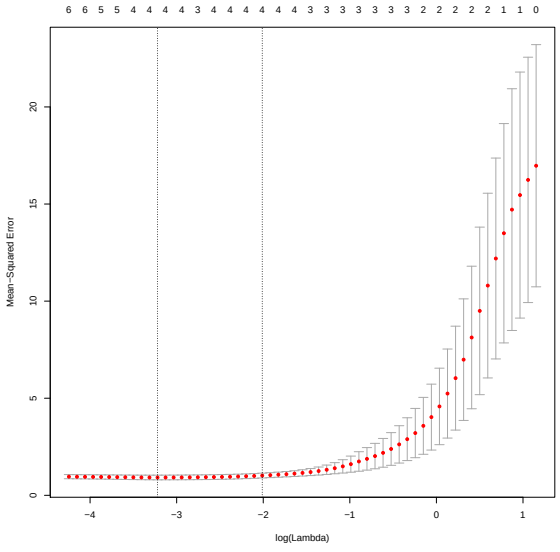
```
> coefficients(model1_fwd, id = 3)
##          (Intercept) poly(x, 10, raw = T)1
##           3.07627412           2.35623596
##poly(x, 10, raw = T)2 poly(x, 10, raw = T)7
##          -3.16514887           0.01046843
> coefficients(model1_bwd, id = 3)
##          (Intercept) poly(x, 10, raw = T)1
##          3.078881355          2.419817953
##poly(x, 10, raw = T)2 poly(x, 10, raw = T)9
##         -3.177235617          0.001870457
```

Forward stepwise picks X^7 over X^3. Backward stepwise with 3 variables picks X^9. Note that $\beta_0 \simeq \hat{\beta}_0$, $\beta_1 \simeq \hat{\beta}_1$ and $\beta_2 \simeq \hat{\beta}_2$.

(e) (5pt) Now fit a lasso model to the simulated data, again using $X, X^2, \ldots, X^{10}$ as predictors. Use cross-validation to select the optimal value of λ. Create plots of the cross-validation error as a function of λ. Report the resulting coefficient estimates, and discuss the results obtained.

Answer:

```
> library(glmnet)
> xmat <- model.matrix(y ~ poly(x, 10, raw = T), data = data1)[, -1]
> mod.lasso <- cv.glmnet(xmat, y, alpha = 1)
> best.lambda <- mod.lasso$lambda.min
> best.lambda
## [1] 0.03991416
> plot(mod.lasso)
> pdf(file="4-e.pdf",width = 10,height = 10)
> plot(mod.lasso)
> dev.off()
> best.model <- glmnet(xmat, y, alpha = 1)
> predict(best.model, s = best.lambda, type = "coefficients")
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                                       1
## (Intercept)             3.0398151056
## poly(x, 10, raw = T)1   2.2303371338
## poly(x, 10, raw = T)2  -3.1033192679
## poly(x, 10, raw = T)3   .
## poly(x, 10, raw = T)4   .
## poly(x, 10, raw = T)5   0.0498410763
## poly(x, 10, raw = T)6   .
## poly(x, 10, raw = T)7   0.0008068431
## poly(x, 10, raw = T)8   .
## poly(x, 10, raw = T)9   .
## poly(x, 10, raw = T)10  .
```



Lasso also picks X^5 over X^3. It also picks X^7 with coefficient equal to 0.0008068.

(f) (5pt) Now generate a response vector Y according to the model

$$Y = \beta_0 + \beta_7 X^7 + \epsilon,$$

and perform best subset selection and the lasso. Discuss the results obtained.

Answer: Create new Y with different $\beta_7 = 7$.

```
> beta_7 <- 7
> Y<- beta_0 + beta_7 * x^7 + eps
> data1 <-data.frame(y = Y, x = x)
> model1 <- regsubsets(y ~ poly(x, 10, raw = T), data = data1, nvmax = 10)
> mod.summary <-summary(model1)
> which.min(mod.summary$cp)
## [1] 2
> which.min(mod.summary$bic)
## [1] 1
> which.max(mod.summary$adjr2)
## [1] 4
> coefficients(model1, id = 1)
##           (Intercept) poly(x, 10, raw = T)7
##               2.95894               7.00077
> coefficients(model1, id = 2)
##           (Intercept) poly(x, 10, raw = T)2
##             3.0704904            -0.1417084
## poly(x, 10, raw = T)7
##             7.0015552
> coefficients(model1, id = 4)
##           (Intercept) poly(x, 10, raw = T)1
##             3.0762524             0.2914016
## poly(x, 10, raw = T)2 poly(x, 10, raw = T)3
##            -0.1617671            -0.2526527
## poly(x, 10, raw = T)7
##             7.0091338
```

We see that BIC picks the most accurate 1-variable model with matching coefficients. Other criteria pick additional variables.

```
> xmat <- model.matrix(y ~ poly(x, 10, raw = T), data = data1)[, -1]
> mod.lasso <-cv.glmnet(xmat, Y, alpha = 1)
> best.lambda <- mod.lasso$lambda.min
> best.lambda
## [1] 13.57478
> best.model <- glmnet(xmat, Y, alpha = 1)
> predict(best.model, s = best.lambda, type = "coefficients")
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                                1
## (Intercept)            3.904188
```

```
## poly(x, 10, raw = T)1  .
## poly(x, 10, raw = T)2  .
## poly(x, 10, raw = T)3  .
## poly(x, 10, raw = T)4  .
## poly(x, 10, raw = T)5  .
## poly(x, 10, raw = T)6  .
## poly(x, 10, raw = T)7  6.776797
## poly(x, 10, raw = T)8  .
## poly(x, 10, raw = T)9  .
## poly(x, 10, raw = T)10 .
```

Lasso also picks the best 1-variable model but intercet is quite off.

P5. In this exercise, we will predict the number of applications received using the other variables in the `college` data set.

(a) (2pt) Split the data set into a training set and a test set.

Answer:

```
> library(ISLR)
> set.seed(11)
> train.size <- nrow(College) / 2
> train <-sample(1:nrow(College), train.size)
> test <- -train
> College.train <- College[train, ]
> College.test <- College[test, ]
```

(b) (3pt) Fit a linear model using least squares on the training set, and report the test error obtained.

Answer:

```
> lm.fit <- lm(Apps~., data=College.train)
> lm.pred <- predict(lm.fit, College.test)
> mean((College.test[, "Apps"] - lm.pred)^2)
## [1] 1538442
```

The test RSS is 1538442.

(c) (5pt) Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.

Answer:

```
> library(glmnet)
> train.mat <- model.matrix(Apps~., data=College.train)
> test.mat <- model.matrix(Apps~., data=College.test)
> grid <-10 ^ seq(4, -2, length=100)
> mod.ridge <- cv.glmnet(train.mat, College.train[, "Apps"], alpha=0, lambda=grid,
```

```
    thresh=1e-12)
> lambda.best <- mod.ridge$lambda.min
> lambda.best
## [1] 18.73817
> ridge.pred <- predict(mod.ridge, newx=test.mat, s=lambda.best)
> mean((College.test[, "Apps"] - ridge.pred)^2)
## [1] 1608859
```

The test RSS is a bit higher than OLS, 1608859.

(d) (5pt) Fit a lasso model on the training set, with λ chosen by cross-validation. Report the test error obtained, along with the number of non-zero coefficient estimates.

Answer:

```
> mod.lasso <- cv.glmnet(train.mat, College.train[, "Apps"], alpha=1, lambda=grid,
    thresh=1e-12)
> lambda.best <- mod.lasso$lambda.min
> lambda.best
## [1] 21.54435
> lasso.pred <- predict(mod.lasso, newx=test.mat, s=lambda.best)
> mean((College.test[, "Apps"] - lasso.pred)^2)
## [1] 1635280
> mod.lasso <- glmnet(model.matrix(Apps~., data=College), College[, "Apps"], alpha=1)
> predict(mod.lasso, s=lambda.best, type="coefficients")
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                           1
## (Intercept) -6.038452e+02
## (Intercept)  .
## PrivateYes  -4.235413e+02
## Accept       1.455236e+00
## Enroll      -2.003696e-01
## Top10perc    3.367640e+01
## Top25perc   -2.403036e+00
## F.Undergrad  .
## P.Undergrad  2.086035e-02
## Outstate    -5.781855e-02
## Room.Board   1.246462e-01
## Books        .
## Personal     1.832912e-05
## PhD         -5.601313e+00
## Terminal    -3.313824e+00
## S.F.Ratio    4.478684e+00
## perc.alumni -9.796600e-01
## Expend       6.967693e-02
## Grad.Rate    5.159652e+00
```

Again here, the test RSS is slightly higher, 1635280.

P6. We have seen that as the number of features used in a model increases, the training error will necessarily decrease, but the test error may not. We will now explore this in a simulated data set.

(a) (2pt) Generate a data set with $p = 20$ features, $n = 1000$ observations, and an associated quantitative response vector generated according to the model

$$Y = X\beta + \epsilon,$$

where β has some elements that are exactly equal to zero.

Answer:

```
> set.seed(1)
> p <-  20
> n <- 1000
> x <-  matrix(rnorm(n * p), n, p)
> B <-  rnorm(p)
> B[3] <-  0
> B[4] <-  0
> B[9] <-  0
> B[19] <- 0
> B[10] <- 0
> eps <- rnorm(p)
> y <- x %*% B + eps
```

(b) (2pt) Split your data set into a training set containing 100 observations and a test set containing 900 observations.

Answer:

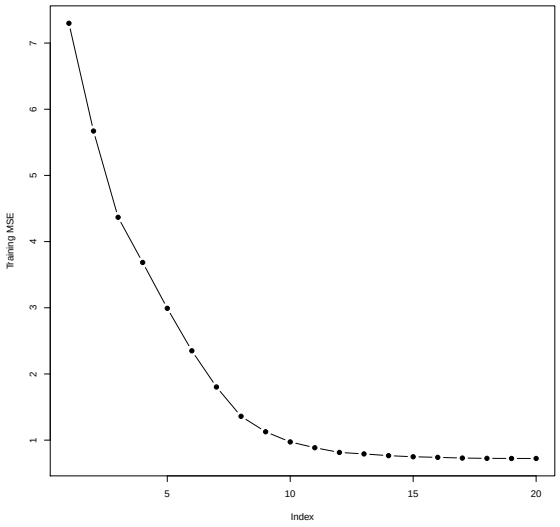
```
> train <- sample(seq(1000), 100, replace = FALSE)
> y.train <- y[train, ]
> y.test <- y[-train, ]
> x.train <- x[train, ]
> x.test <- x[-train, ]
```

(c) (2pt) Perform best subset selection on the training set, and plot the training set MSE associated with the best model of each size.

Answer:

```
> library(leaps)
> regfit.full<- regsubsets(y ~ ., data = data.frame(x = x.train, y = y.train),
+                          nvmax = p)
> val.errors <- rep(NA, p)
> x_cols <- colnames(x, do.NULL = FALSE, prefix = "x.")
> for (i in 1:p) {
+   coefi <- coef(regfit.full, id = i)
+   pred <-as.matrix(x.train[, x_cols %in% names(coefi)]) %*% coefi[names(coefi) %in%
+                                                                   x_cols]
```

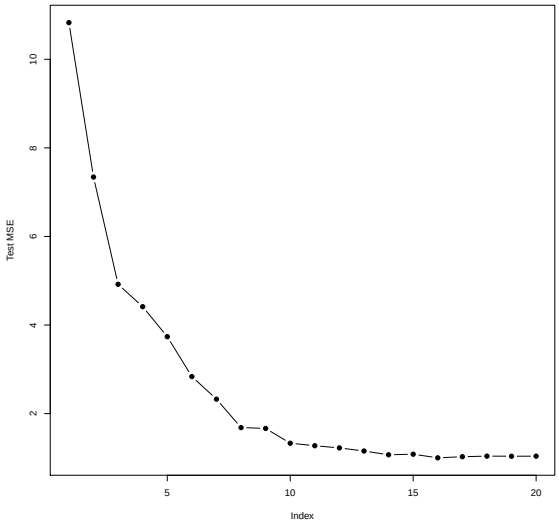
```
+   val.errors[i] <- mean((y.train - pred)^2)
+ }
> pdf(file="6-c.pdf",width = 10,height = 10)
> plot(val.errors, ylab = "Training MSE", type = "b",pch=19)
> dev.off()
```



(d) (2pt) Plot the test set MSE associated with the best model of each size.

Answer:

```
> val.errors <- rep(NA, p)
> for (i in 1:p) {
+   coefi <- coef(regfit.full, id = i)
+   pred <- as.matrix(x.test[, x_cols %in% names(coefi)]) %*% coefi[names(coefi) %in%
+                                                                   x_cols]
+   val.errors[i] <- mean((y.test - pred)^2)
+ }
> pdf(file="6-d.pdf",width = 10,height = 10)
> plot(val.errors, ylab = "Test MSE", type = "b",pch=19)
> dev.off()
```



(e) (2pt) For which model size does the test set MSE take on its minimum value? Comment on your results. If it takes on its minimum value for a model containing only an intercept or a model containing all of the features, then play around with the way that you are generating the data in (a) until you come up with a scenario in which the test set MSE is minimized for an intermediate model size.

Answer:

```
> which.min(val.errors)
## [1] 16
```

16-parameter model has the smallest test MSE.

(f) (2pt) How does the model at which the test set MSE is minimized compare to the true model used to generate the data? Comment on the coefficient values.

Answer:

```
> coef(regfit.full, id = 16)
## (Intercept)         x.1         x.2         x.5         x.6         x.7         x.8        x.11
##  0.09613244  0.28256751  0.19385802  0.99994674 -0.28597795 -1.50482273  0.77817125  0.90815918
##        x.12        x.13        x.14        x.15        x.16        x.17        x.18        x.19
##  0.48477881 -0.19747066 -0.71978955 -0.74282068 -0.33900837  0.12234642  1.74270174 -0.12435131
##        x.20
## -1.03003019
```

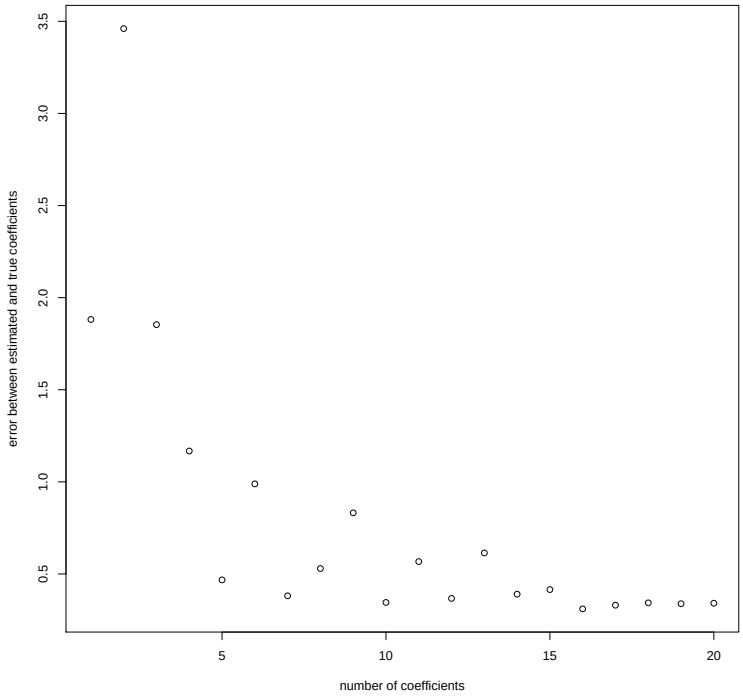
We got all the coefficients not equal to zero.

(g) (3pt) Create a plot displaying $\sqrt{\sum_{j=1}^{p}(\beta_j - \hat{\beta}_j^r)^2}$ for a range of values of r, where $\hat{\beta}_j^r$ is the jth coefficient estimate for the best model containing r coefficients. Comment on what you observe. How does this compare to the test MSE plot from (d)?

Answer:

```
> val.errors <- rep(NA, p)
> a <- rep(NA, p)
```

```
> b <-  rep(NA, p)
> for (i in 1:p) {
+   coefi <-  coef(regfit.full, id = i)
+   a[i] <- length(coefi) - 1
+   b[i] <- sqrt(sum((B[x_cols %in% names(coefi)] - coefi[names(coefi) %in% x_cols])^2) +
+                sum(B[!(x_cols %in% names(coefi))]^2))
+ }
> pdf(file="6-f.pdf",width = 10,height = 10)
> plot(x = a, y = b, xlab = "number of coefficients",
       ylab = "error between estimated and true coefficients")
> dev.off()
```



The error quickly drops as r increases to 5, and then levels off at around 15. The plot is qualitatively similar to the MSE plot in (d).