Lab 4

In the hot New Jersey night

Alyssa Andrichik

10/7/2019

```
d <- read.csv("http://andrewpbray.github.io/data/crime-train.csv")
library(glmnet)
library(tidyverse)
d[d=="?"]<-NA
d[d==""]<-NA
real_d <- na.omit(d)
```

Ridge Regression Model

```
X <- model.matrix(ViolentCrimesPerPop ~ ., data = real_d)[, -1]
X[1:2, ]
Y <- real_d$ViolentCrimesPerPop
lambdas <- seq(from = 1e4, to = 1e-2, length.out = 100)
ridge_mod <- glmnet(x = X, y = Y, alpha = 0,
              lambda = lambdas, standardize = TRUE)

set.seed(1)
training = sample(1:nrow(X), nrow(X)/2)
testing = (-training)
y_test = Y[testing]

cv_ridge_mod <- cv.glmnet(X[training,], Y[training],
                          alpha = 0, lambda = lambdas)
plot(cv_ridge_mod)

opt_lambda <- cv_ridge_mod$lambda.min
opt_lambda

#MSE
set.seed(1)
training = sample(1:nrow(X), nrow(X)/2)
testing = (-training)
y_test = Y[testing]

ridge_mod2 = glmnet(X[training,], Y[training], alpha = 0,
                    lambda = lambdas, thresh = 1e-12)
ridge_predict = predict(ridge_mod2, s = opt_lambda,
                        newx = X[testing,])
mean((ridge_predict - y_test)^2)

out = glmnet(X, Y, alpha = 0)
predict(out, type = "coefficients", s = opt_lambda) [1:52,]
```

LASSO Model

```
X <- model.matrix(ViolentCrimesPerPop ~ ., data = real_d)[, -1]
X[1:2, ]
Y2 <- real_d$ViolentCrimesPerPop
lambdas <- seq(from = 1e4, to = 1e-2, length.out = 100)
lasso_mod <- glmnet(x = X, y = Y2, alpha = 1,
              lambda = lambdas, standardize = TRUE)
set.seed(1)
training = sample(1:nrow(X), nrow(X)/2)
testing = (-training)
y2_test = Y2[testing]

cv_lasso_mod <- cv.glmnet(X[training,],
                          Y2[training], alpha = 1, lambda = lambdas)
plot(cv_lasso_mod)

opt2_lambda <- cv_lasso_mod$lambda.min
opt2_lambda

#MSE
set.seed(1)
training = sample(1:nrow(X), nrow(X)/2)
testing = (-training)
y2_test = Y2[testing]

lasso_mod2 = glmnet(X[training,], Y2[training], alpha = 1,
                    lambda = lambdas, thresh = 1e-12)
lasso_mod2
lasso_predict = predict(lasso_mod2, s = opt2_lambda, newx = X[testing,])
mean((lasso_predict - y2_test)^2)

out2 = glmnet(X, Y2, alpha = 1)
predict(out2, type = "coefficients", s = opt2_lambda) [1:12,]
```

1)How many variables were selected by the LASSO?

12 out of the 52 I used in the ridge regression.

2)What are the training MSEs for ridge and LASSO using the optimal value of (λ)?

The training MSE for ridge, using the optimal value of λ, is 0.06045228 The training MSE for LASSO, using the optimal value of λ, is 0.06972382

3)If the MSEs differed, why do you think one is higher than the other in this setting?

LASSO has a major advantage over ridge regression since it creates a more sparse model that only involves a subset of the variables, this is because the model shrinks the coeffcient estimates to be exactly zero to remove the variables that have a tuning parameter λ that is sufficently large, while ridge regression includes all predictors in the final model. LASSO is a simpler and more interpretable model, but that does not mean that it has better prediction accuracy. The MSE for the ridge regression is lower than the MSE for LASSO. This means that the average squared difference between the estimated values and the actual value for the model

with less variables is higher, and using all the parameters available was actually more accurate than the variable selection LASSO did. LASSO overcompensated and did not account that some of the predictors it removed were related to the response and would have helped create a more accurate model. Ridge regression is more flexible than LASSO which made the model more accurate, but the LASSO model is far more easier to interpret.

Problem Set 2

2.

(a) The lasso, relative to least squares is

Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance (iii). The lasso is a more restrictive model since it decreases the number of variables, and thus has less varience in predictions and more bias due to the increase in constraints. The goal of a lasso model is to reduce the chance of overfitting a model, so it would work better than least squares which is likely to overfit a model since it uses more parameters that might not be necessary.

(b) Ridge regression, relative to least squares is

Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance (iii). iii is the correct answer again. A ridge regression is more restrictive than least squares for the same reasons (increased bias and decrease in varience), but is more flexible than the lasso.

3. Suppose we estimate the regression coefficients in a linear regression model by minimizing

$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}) subject\ to \sum_{j=1}^{p}|\beta_j| \leq s$

for a particular value of s.

(a) As we increase s from 0, the training RSS will:

Steadily decrease (iv). Increasing the s makes the model more flexible since the restriction on β is reducing. This causes the RSS to decrease.

(b) As we increase s from 0, the test RSS will:

Decrease initially, and then eventually start increasing in a U shape (ii). As we increase s from 0, the model becomes more and more flexible since the β is being restricted less and less. So, as more and more coefficents are forced to 0, the test RSS will decrease initially and improve the model. But then, as necessary coefficents are removed from the model, the RSS will increase again making a U shape.

(c) As we increase s from 0, the variance will:

Steadily increase (iii). As we increase s from 0, the model becomes more and more flexible since the β is being restricted less and less. Variance steadily increases with the increase in model flexibility since there are more predictors addressing the response.

(d) As we increase s from 0, the (squared) bias will:

Steadily decrease (iv). As we increase s from 0, the model becomes more and more flexible since the β is being restricted less and less. Bias decreases with the increase in the model flexibility since there are more predictors addressing the response.

(e) As we increase s from 0, the irreducible error will:

Remain constant (v). Irreducible error is independant of the model, and thus is also independant of the value of s.

4. Suppose we estimate the regression coefficients in a linear regression model by minimizing

$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}) + \lambda\sum_{j=1}^{p}\beta_j^2$

for a particular value of s.

(a) As we increase λ from 0, the training RSS will:

Steadily increase (iii). Increasing the λ makes the model less flexible since there is more restriction on β. This will lead to an increase in the training RSS.

(b) As we increase λ from 0, the test RSS will:

Decrease initially, and then eventually start increasing in a U shape (ii). As we increase λ from 0, the model becomes less and less flexible since the β is being restricted more and more. This causes a decrease at first in the test RSS, but will increase again into the usual U shape

(c) As we increase λ from 0, the variance will:

Steadily decrease (iv). Increasing the λ makes the model less flexible since there is more restriction on the β. Variance steadily decreases with the decrease in model flexibility since there are less predictors addressing the response.

(d) As we increase λ from 0, the (squared) bias will:

Steadily increase (iii). As we increase λ from 0, the model becomes less flexible since there is more restriction on the β. Bias increases with the decrease in the model flexibility since there are less predictors addressing the response.

(e) As we increase λ from 0, the irreducible error will:

Remain constant (v). Irreducible error is independant of the model, and thus is also independant of the value of λ.

6. We will now explore (6.12) and (6.13) further.

6.12: $\sum_{j=1}^{p}(y_j - \beta_j)^2 + \alpha\sum_{j=1}^{p}\beta_j^2$

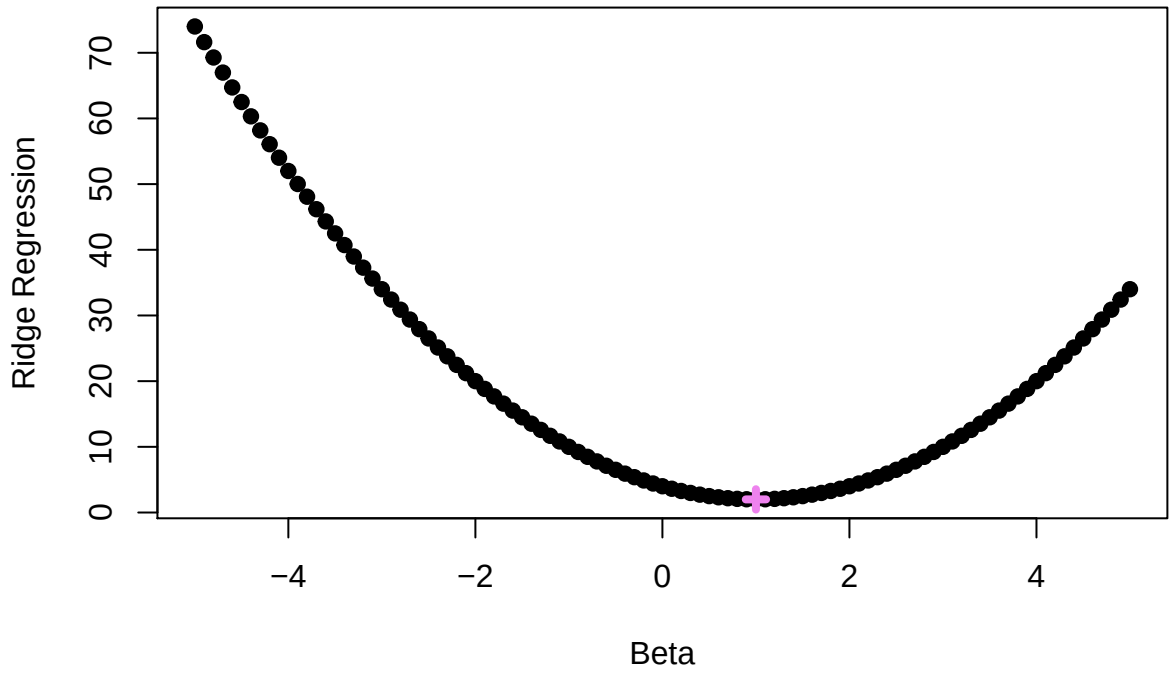
6.13: $\sum_{j=1}^{p}(y_j - \beta_j)^2 + \alpha\sum_{j=1}|\beta_j|$

6.14: $\hat{\beta}_j^R = \frac{y_j}{1+\alpha}$

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

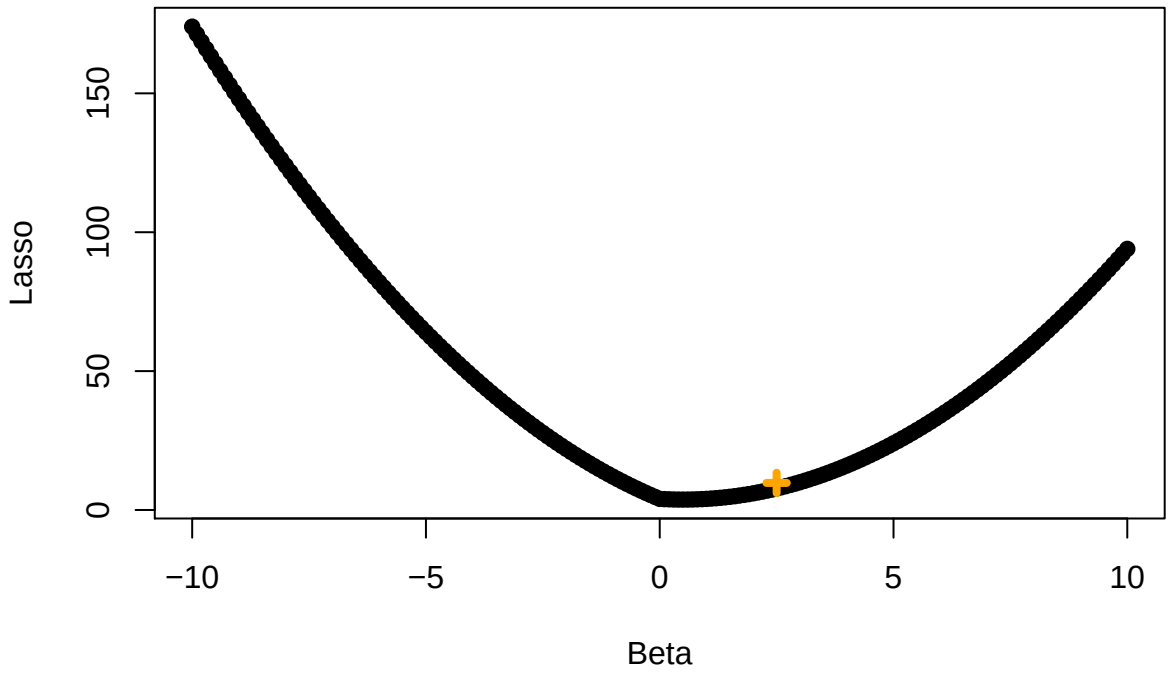
(a) Consider (6.12) with $p = 1$. For some choice of y_1, x_1, and $\alpha > 0$, plot (6.12) as a function of β_1. Your plot should confirm that (6.12) is solved by (6.14).

```
y <- 2
lambda1 <- 1
beta <- seq(-5, 5, 0.1)
plot1<- plot(beta, ((y-beta)^2+lambda1*beta^2), pch = 19,
             xlab = "Beta", ylab = "Ridge Regression")
beta_point <- y/(1+lambda1)
points(beta_point, (y-beta_point)^2+lambda1 * beta_point^2,
       pch = 3, col = "violet", lwd = 4)
```



(b) Consider (6.13) with $p = 1$. For some choice of y_1, x_1, and $\alpha > 0$, plot (6.13) as a function of β_1. Your plot should confirm that (6.13) is solved by (6.15).

```
y_2<- 4
lambda_2<- 3
beta_2<- seq(-10,10,0.1)
plot2<- plot(beta_2, (y-beta_2)^2+lambda_2*abs(beta_2), pch = 19,
             xlab = "Beta" , ylab = "Lasso")
beta2_est <- y_2 - ((lambda_2)/2)
points(beta2_est, (y_2 - beta2_est)^2 + lambda_2 * abs(beta2_est),
       col = "orange", pch = 3, lwd = 4)
```


Lasso
0
50
100
150
−10
−5
0
5
10
Beta