Stats 101C HW 4

Charles Liu (304804942)

11/10/2020

Loading Necessary Packages:

```
library(ISLR)
library(pls)
```

```
##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings
```

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.0-2
```

Problem 1 (Exercise 6.8.2)

For parts (a) through (c), indicate which of i. through iv. is correct. Justify your answer.

(a) The lasso, relative to least squares, is:

ANSWER: (iii) is correct because LASSO shrinks the coefficient estimates to zero. This will cause our model to become less flexible, having to fit to zero, and it will be removing "non-essential" predictors to the model. From the Variance-Bias Tradeoff, we know that with a less flexibility, we will have a decreased variance with an increased bias. One important thing to note is that this depends on the penalty factor, as having a higher penalty means increased bias and decreased variance.

(b) Repeat (a) for ridge regression relative to least squares.

ANSWER: (iii) is correct because Ridge Regression shrinks the coefficient estimates to as "*close*" to zero as possible. This will cause our model to become less flexible, having to fit closer to zero, and it will be both removing and keeping some "non-essential" predictors to the model. From the Variance-Bias Tradeoff, we know that with a less flexibility, we will have a decreased variance with an increased bias. One important thing to note is that this depends on the penalty factor, as having a higher penalty means increased bias and decreased variance.

(c) Repeat (a) for non-linear methods relative to least squares.

ANSWER: (ii) is correct because Non-Linear Methods are usually more flexible than Least Squares to match the model. However, Non-Linear Methods will have a decreased bias and an increased variance due to it being more flexible.

CHOICES:

i) More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

ii) More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

iii) Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

iv) Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

Problem 2 (Exercise 6.8.5)

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) Write out the ridge regression optimization problem in this setting.

ANSWER: $(y_1 - \hat{\beta_1} x_1 - \hat{\beta_2} x_1)^2 + (y_2 - \hat{\beta_1} x_2 - \hat{\beta_2} x_2)^2 + \lambda(\hat{\beta_1}^2 + \hat{\beta_2}^2)$ because we know that $x_1 = x_{11} = x_{12}$ & $x_2 = x_{21} = x_{22}$.

(b) Argue that in this setting, the ridge coefficient estimates satisfy $\hat{\beta_1} = \hat{\beta_2}$.

ANSWER: To satisfy the argument above, we would start by taking the derivative of $\hat{\beta_1}$ and $\hat{\beta_2}$ and set them equal to zero, respectively to get the following equations below...

$y_1 x_1 + y_2 x_2 = \hat{\beta_1}(x_1^2 + x_2^2 + \lambda) + \hat{\beta_2}(x_1^2 + x_2^2)$

$y_1 x_1 + y_2 x_2 = \hat{\beta_1}(x_1^2 + x_2^2) + \hat{\beta_2}(x_1^2 + x_2^2 + \lambda)$

After that, we would set the two derivatives equal to each other and simplify the equations to get $\hat{\beta_1} = \hat{\beta_2}$.

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

ANSWER: This is similar to (5a), but the only difference is that the $\hat{\beta}'s$ will be in absolute value rather than squared. $(y_1 - \hat{\beta_1} x_1 - \hat{\beta_2} x_1)^2 + (y_2 - \hat{\beta_1} x_2 - \hat{\beta_2} x_2)^2 + \lambda(|\hat{\beta_1}| + |\hat{\beta_2}|)$ because we know that $x_1 = x_{11} = x_{12}$ & $x_2 = x_{21} = x_{22}$.

(d) 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: We have the constraint subject to $|\hat{\beta}_1| + |\hat{\beta}_2| \leq g$ and have to minimize the equations below (both work) based on the given equations from the question…

$0 \leq 2(y_1 - (\hat{\beta}_1 + \hat{\beta}_2)x_1)^2$ $(\hat{\beta}_1 + \hat{\beta}_2) = \frac{y_1}{x_1}$ (minimized)

$0 \leq 2(y_2 - (\hat{\beta}_1 + \hat{\beta}_2)x_2)^2$ $(\hat{\beta}_1 + \hat{\beta}_2) = \frac{y_2}{x_2}$ (minimized)

We find that $g = \frac{y_1}{x_1} = \frac{y_2}{x_2}$. With all these equations, we can see that $\hat{\beta}_1 + \hat{\beta}_2 = \pm g$ and have $\hat{\beta}_1, \hat{\beta}_2 \geq 0$ (if g is positive) and $\hat{\beta}_1, \hat{\beta}_2 \leq 0$ (if g is negative). Therefore, we have proven that $\hat{\beta}_1$ and $\hat{\beta}_2$ are NOT unique and have many possible solutions based on what is stated above.

Problem 3 (Exercise 6.8.9)

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

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

```
# Load data and set.seed(...) for sampling
data(College)
set.seed(1)

# 50% of data for train and 50% of data for test
train_size <- floor(0.5 * nrow(College))
train_ind <- sample(seq_len(nrow(College)), size = train_size)
college_train <- College[train_ind, ]
college_test <- College[-train_ind, ]

# Create our x and y training and test
x_train <- model.matrix(Apps~., college_train)[,-1]
y_train <- college_train$Apps
x_test <- model.matrix(Apps~., college_test)[,-1]
y_test <- college_test$Apps
```

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

```
college_lm <- lm(Apps ~ ., data = college_train)
pred_lm <- predict(college_lm, college_test)
test_error_lm <- mean((pred_lm - y_test)^2)
test_error_lm
```

```
## [1] 1135758
```

ANSWER: The test error for Linear Model is 1135758.

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

```
college_ridge <- glmnet(x_train, y_train, alpha=0)
cv_college_ridge <- cv.glmnet(x_train, y_train, alpha=0)
best_lambda_ridge <- cv_college_ridge$lambda.min
best_lambda_ridge
```

```
## [1] 405.8404
```

```r
pred_ridge <- predict(college_ridge, s=best_lambda_ridge, newx=x_test)
test_error_ridge <- mean((pred_ridge - y_test)^2)
test_error_ridge
```

```
## [1] 976261.5
```

ANSWER: The test error for Ridge Regression Model is 976261.5 with $\lambda = 405.8404$.

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

```r
college_lasso <- glmnet(x_train, y_train, alpha=1)
cv_college_lasso <- cv.glmnet(x_train, y_train, alpha=1)
best_lambda_lasso <- cv_college_lasso$lambda.min
best_lambda_lasso
```

```
## [1] 1.97344
```

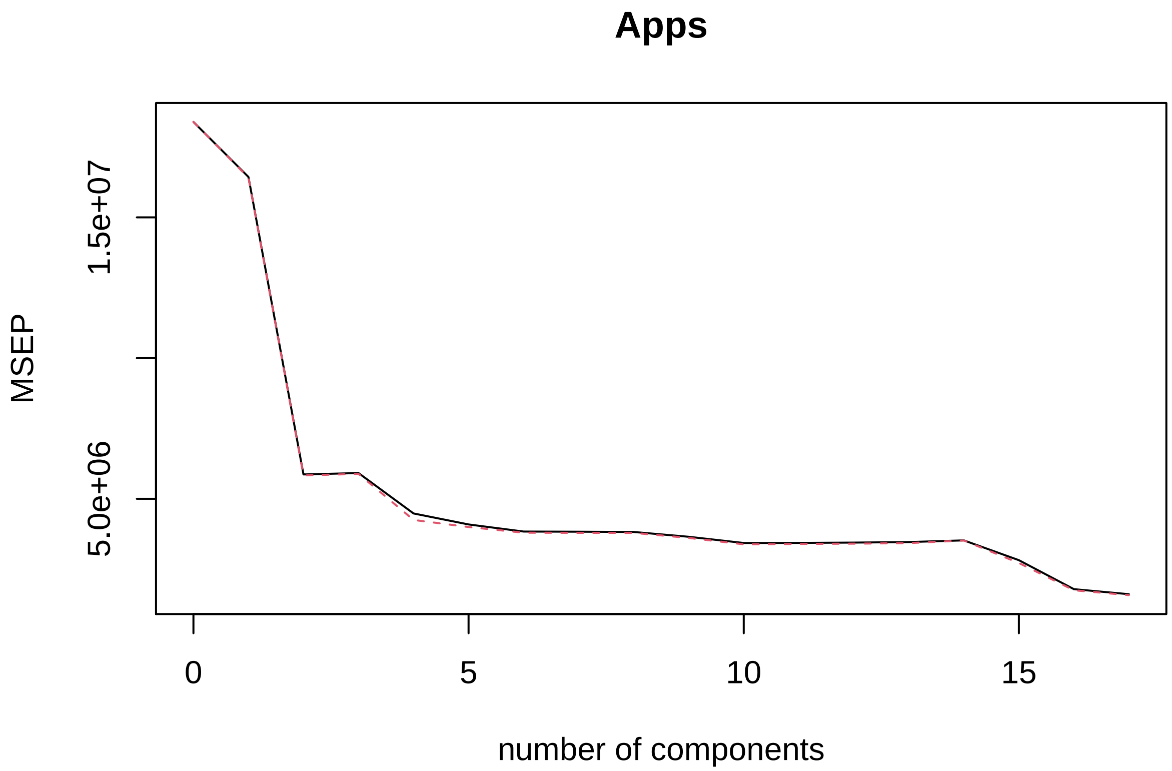
```r
pred_lasso <- predict(college_lasso, s=best_lambda_lasso, newx=x_test)
test_error_lasso <- mean((pred_lasso - y_test)^2)
test_error_lasso
```

```
## [1] 1115901
```

ANSWER: The test error for LASSO Model is 1115901 with $\lambda = 1.97344$.

(e) Fit a PCR model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.

```r
college_pcr <- pcr(Apps~., data= college_train, scale=TRUE, validation="CV")
validationplot(college_pcr, val.type = 'MSEP') # PCR Plot
```



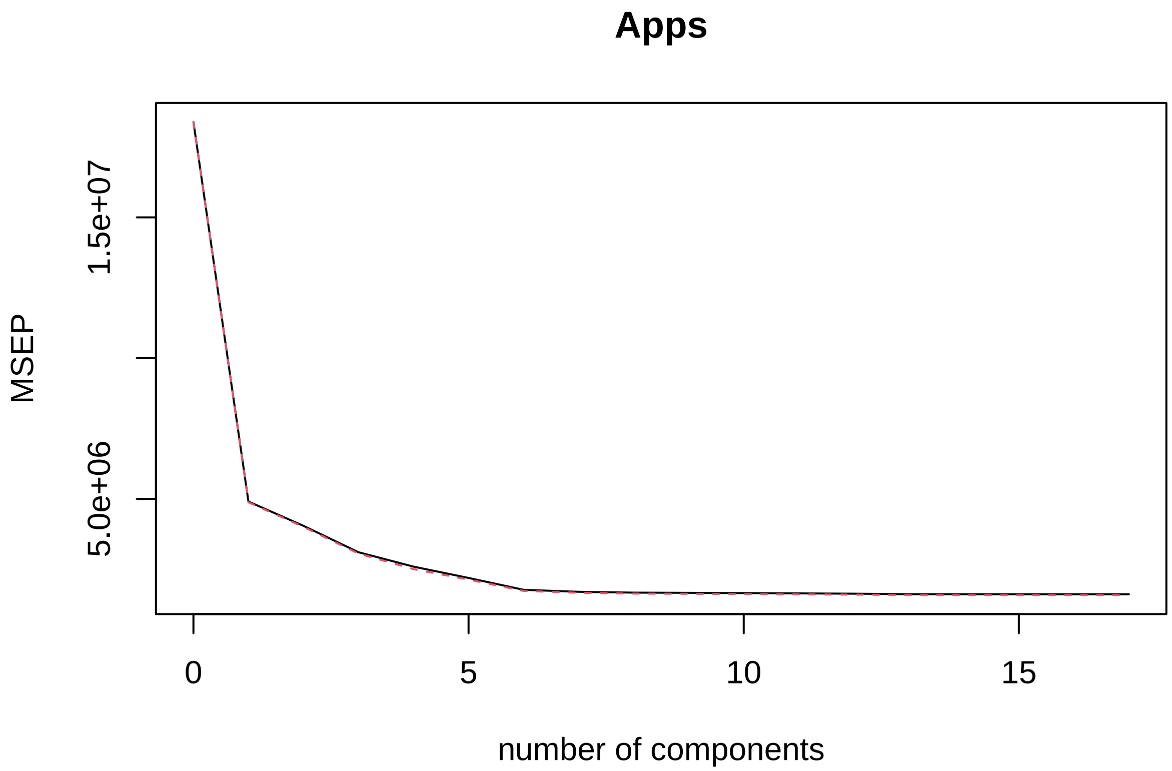
```
pred_pcr <- predict(college_pcr, x_test)
test_error_pcr <- mean((pred_pcr - y_test)^2)
test_error_pcr
```

```
## [1] 2403120
```

ANSWER: The test error for PCR Model is 2403120.

(f) Fit a PLS model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.

```
college_plsr <- plsr(Apps~., data=college_train,
                     scale=TRUE, validation="CV")
validationplot(college_plsr, val.type="MSEP") # PLSR Plot
```



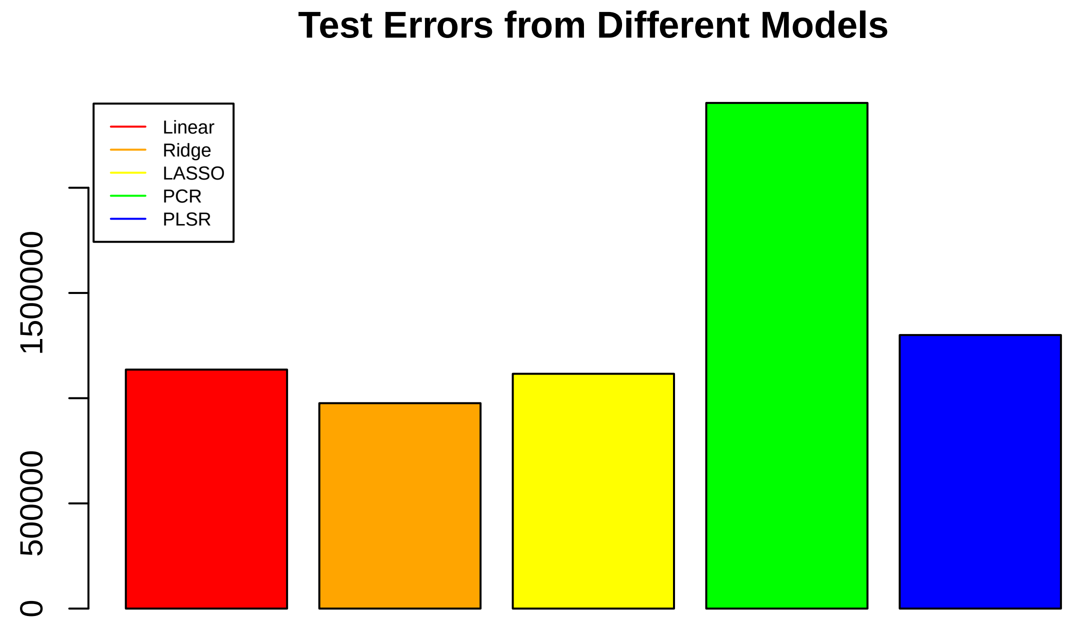
```
pred_plsr <- predict(college_plsr, x_test)
test_error_plsr <- mean((pred_plsr - y_test)^2)
test_error_plsr
```

```
## [1] 1300038
```

ANSWER: The test error for PLSR Model is 1300038.

(g) Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these five approaches?

```
# Plot Comparison
barplot(c(test_error_lm, test_error_ridge, test_error_lasso,
          test_error_pcr, test_error_plsr),
        col=c("red", "orange", "yellow", "green", "blue"),
        main="Test Errors from Different Models")
legend(0, 2400000, legend = c("Linear", "Ridge", "LASSO", "PCR", "PLSR"),
       col=c("red", "orange", "yellow", "green", "blue"),
       lty = c(1,1,1,1,1), cex = 0.6)
```



```
# Numerical Comparison
test_error_matrix <- matrix(c(test_error_lm, test_error_ridge,
                              test_error_lasso, test_error_pcr,
                              test_error_plsr))
colnames(test_error_matrix) <- "MSE"
rownames(test_error_matrix) <- c("Linear Model", "Ridge Regression",
                                 "LASSO", "PCR", "PLSR")
test_error_matrix
```

```
##                        MSE
## Linear Model     1135758.3
## Ridge Regression  976261.5
## LASSO            1115900.6
## PCR              2403119.9
## PLSR             1300037.9
```

COMMENTS: When we compare the Test Errors, we can see that Ridge Regression provides the lowest test errors, which means this method might be the best model to use for this situation. We can see that PCR has the highest amount of Test Errors, meaning we should avoid using PCR model for this data. Linear and LASSO methods both have approximately the same MSE (approximately by 20000 difference). PLSR method provides the second highest MSE, which tells us we should use other methods rather than PLSR.