Computational Statistics II

Lab 6 Homework

Subash Kharel

April 17, 2020

1. **(Question 9, Page 263) 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.**

```
library(ISLR)

set.seed(1)
train = sample(c(TRUE, FALSE), nrow(College), rep=TRUE)
test = (!train)
```

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

```
regfit.best = regsubsets(Apps~., data=College[train,], nvmax = 18)
test.mat = model.matrix(Apps~., data = College[test,])

# Vector to store errors for different models
val.errors = rep(NA, 18)

for(i in 1:18){
  coefi = coef(regfit.best, id = i)
  pred = test.mat[, names(coefi)]%*%coefi
  val.errors[i] = mean((College$Apps[test]-pred)^2)
}

# Determining the minimum value for errors in vector
min_error = which.min(val.errors)
cat("Error using linear regression is ", val.errors[min_error])
```

```
Error using linear regression is  1407758
```

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

```
x = model.matrix(Apps~., College)[,-1]
y = College$Apps

# Getting the best lambda using cross validation
cv.out = cv.glmnet(x[train,], y[train], alpha = 0)
best_lambda = cv.out$lambda.min

grid=10^seq(10,-2, length =50)
ridge.mod = glmnet(x[train,], y[train], alpha = 0, lambda = best_lambda)
ridge.pred = predict(ridge.mod, s= best_lambda, newx = x[test,])
error = mean((ridge.pred - y[test])^2)
cat("Error using ridge regression is", error)
```

```
Error using ridge regression is 2293643
```

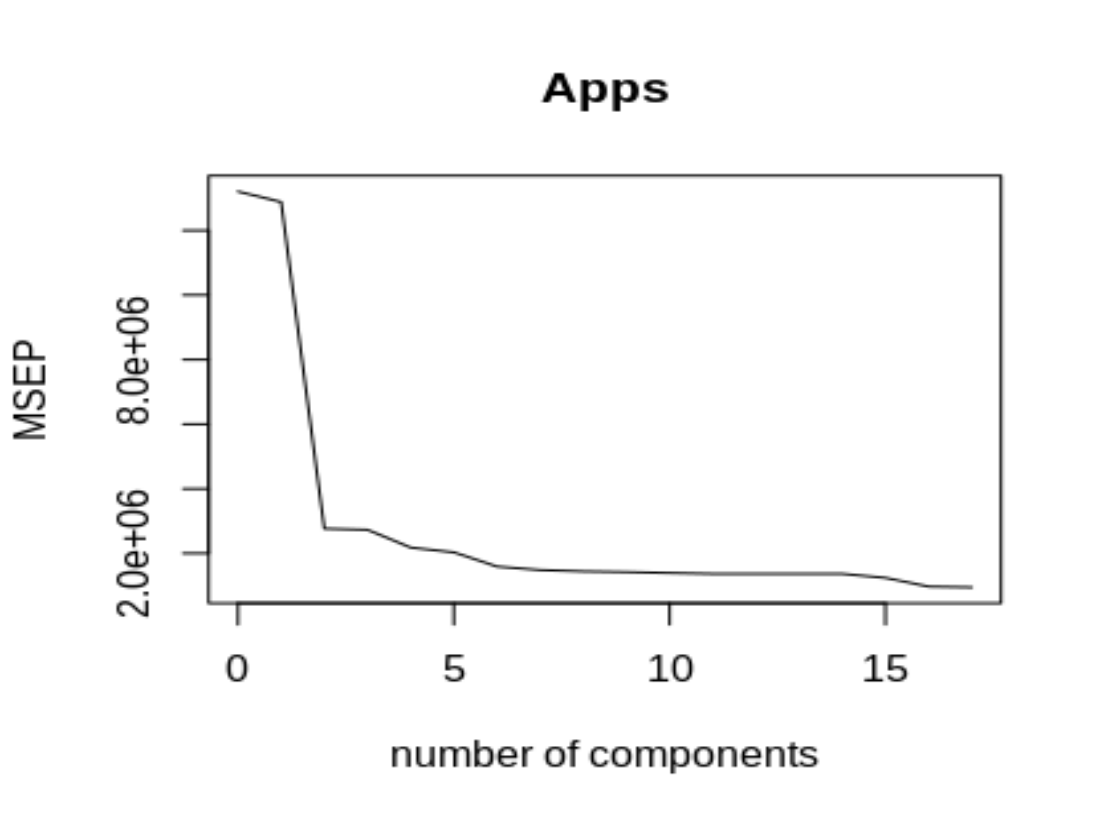
d. **Fit a lasso model on the training set, with λ chosen by cross validation. Report the test error obtained, along with the number of non-zero coefficient estimates.**

```
cv.out = cv.glmnet(x[train,], y[train], alpha = 1)
best_lambda = cv.out$lambda.min
lasso.mod = glmnet(x[train,], y[train], alpha = 1, lambda = best_lambda)
out = glmnet(x, y, alpha = 1, lambda = best_lambda)
lasso.coef = predict(out, type="coefficients", s= best_lambda)[1:18,]
num_of_non_zero_coefficients = length(lasso.coef[lasso.coef != 0])
lasso.pred = predict(lasso.mod, s=best_lambda, newx=x[test,])
error = mean((lasso.pred-y[test])^2)
cat("Number of non-zero coefficients is ", num_of_non_zero_coefficients)
cat("Error using lasso regression is", error)
```

```
Number of non-zero coefficients is  16
Error using lasso regression is 1492700
```

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

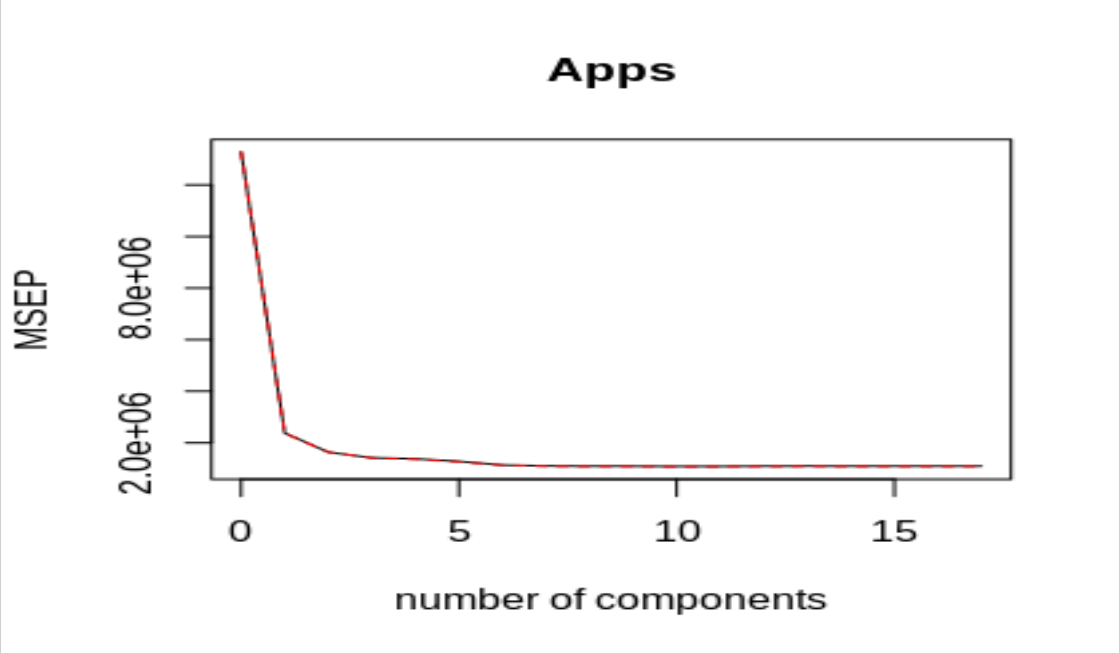
```
library(pls)
pcr.fit = pcr(Apps~., data=College, subset = train, scale=TRUE, valication="CV")
validationplot(pcr.fit, val.type="MSEP")
# From validation plot, we see that the lowest cross validation error occurs when M = 17 (largest)
pcr.pred = predict(pcr.fit, x[test,], ncomp = 17)
error = mean ((pcr.pred-y[test])^2)
cat("Error using pcr is", error)
```



Error using pcr is 1454250

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

```
pls.fit = plsr(Apps~., data = College, subset = train, scale = TRUE, validation="CV")
validationplot(pls.fit, val.type="MSEP")
# From validation plot, we see that the lowest cross validation error occurs when M = 17
pls.pred = predict(pls.fit, x[test,], ncomp = 17)
error = mean((pls.pred-y[test])^2)
cat("Error using pcr is", error)
```



Error using pcr is 1454250

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?**
Answer: From the above experiments, we get the following results:

Regression Method	Mean Squared Error
Linear Regression	**1407758**
Ridge Regression	**2293643**
Lasso Regression	**1492700**
PCR	**1454250**
PLS	**1454250**

We can see that the Linear regression gives the best approximation to the test data. Both the PCR and PLS have the same accuracy, giving the test result of 1454250. Ridge regression appears to be the last on the list giving the highest amount of test error.

Using the Linear regression, we can predict the number of college application with best accuracy compared to other methods