Chapter 6 HW

Ulices Gonzalez

April 24th, 2025

## Conceptual Questions

### 1. We perform best subset, forward step wise, and backward step wise selection on a single data set. For each approach, we obtain p + 1 models, containing 0, 1, 2,…,p predictors. Explain your answers:

#### (a) Which of the three models with k predictors has the smallest training RSS?

Best subset. since it takes all possibilities with k predictors

#### (b) Which of the three models with k predictors has the smallest test RSS?

There isn’t really a definitive choice since the value of k is what makes the determination but given that best subset looks at all possibilities it would likely be the best choice

#### (c) True or False:

##### i. The predictors in the k-variable model identified by forward step wise are a subset of the predictors in the (k+1)-variable model identified by forward step wise selection

True - The model would have an extra predictor but otherwise the same as the k variable

##### ii. The predictors in the k-variable model identified by backward step wise are a subset of the predictors in the (k + 1)- variable model identified by backward step wise selection.

True - The model would have one less predictor but otherwise the same as the k variable

##### iii. The predictors in the k-variable model identified by backward step wise are a subset of the predictors in the (k + 1)- variable model identified by forward step wise selection.

False - The forward and backward have two different points at which they begin

##### iv. The predictors in the k-variable model identified by forward step wise are a subset of the predictors in the (k+1)-variable model identified by backward step wise selection.

False - Similar to iii the forward and backward models have different starting points

##### v. The predictors in the k-variable model identified by best subset are a subset of the predictors in the (k + 1)-variable model identified by best subset selection.

False - There is no way for this to occur definitively as there are too many possibilities identified by the best subset model

### 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:

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

The lasso shrinks variables to zero making it simpler and less flexible by definition

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

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

Similar to the lasso it shrinks variables but to near zero making less flexible as well

#### (c) Non-linear methods relative to least squares:

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

Most non linear methods like the polynomial regression can handle more predictors/complexity making it more flexible and other methods similar to it are similarly flexible

## Applied Questions

### 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.

library(glmnet)

library(pls)

##   
## Attaching package: 'pls'

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

college <- read.csv("DataSets/College.csv")  
college <- college[, -1]  
college$Private <- as.factor(college$Private)  
  
set.seed(3)  
samples <- sample(1:nrow(college), 0.6 \* nrow(college))  
trainingData <- college[samples, ]  
testData <- college[-samples, ]

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

collegeLm <- lm(Apps ~ ., data = trainingData)  
collegePredict <- predict(collegeLm, testData)  
collegeMSE <- mean((testData$Apps - collegePredict)^2)  
  
print(collegeMSE)

## [1] 1341249

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

trainingX <- model.matrix(Apps ~ ., data = trainingData)[, -1]  
trainingY <- trainingData$Apps  
testX <- model.matrix(Apps ~ ., data = testData)[, -1]  
testY <- testData$Apps  
  
cvRidge <- cv.glmnet(trainingX, trainingY, alpha = 0, nfolds = 10)  
bestLambda <- cvRidge$lambda.min  
  
collegeRidge <- glmnet(trainingX, trainingY, alpha = 0, lambda = bestLambda)  
  
ridgePredict <- predict(collegeRidge, s = bestLambda, newx = testX)  
  
ridgeMSE <- mean((testY - ridgePredict)^2)  
  
print(ridgeMSE)

## [1] 2169678

#### (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.

cvLasso <- cv.glmnet(trainingX, trainingY, alpha = 1, nfolds = 10)  
bestLambda <- cvLasso$lambda.min  
  
collegeLasso <- glmnet(trainingX, trainingY, alpha = 1, lambda = bestLambda)  
lassoPredict <- predict(collegeLasso, s = bestLambda, newx = testX)  
lassoMSE <- mean((testY - lassoPredict)^2)  
nonZeros <- sum(coef(collegeLasso, s = bestLambda) != 0) - 1  
  
print(lassoMSE)

## [1] 1304991

print(nonZeros)

## [1] 14

#### (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.

cvPCR <- plsr(Apps ~ ., data = trainingData, scale = TRUE, validation = "CV", segments = 10)  
  
optimalM <- which.min(RMSEP(cvPCR)$val[1, , ])  
collegePCR <- pcr(Apps ~ ., data = trainingData, scale = TRUE, ncomp = optimalM)  
pcrPredict <- predict(collegePCR, newdata = testData, ncomp = optimalM)  
pcrMSE <- mean((testData$Apps - pcrPredict)^2)  
  
print(pcrMSE)

## [1] 2980770

print(optimalM)

## 9 comps   
## 10

#### (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.

cvPLS <- plsr(Apps ~ ., data = trainingData, scale = TRUE, validation = "CV", segments = 10)  
  
optimalM <- which.min(RMSEP(cvPLS)$val[1, , ])  
collegePLS <- plsr(Apps ~ ., data = trainingData, scale = TRUE, ncomp = optimalM)  
plsPredict <- predict(collegePLS, newdata = testData, ncomp = optimalM)  
plsMSE <- mean((testData$Apps - plsPredict)^2)  
  
print(plsMSE)

## [1] 1347468

print(optimalM)

## 10 comps   
## 11

#### (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 fve approaches?

All the models seem to have a relatively high MSE which isn’t the most ideal in regards to predictions. The lasso regression performed the best as it has the lowest of the MSE’s reported. The linear regression and PLS performed similarly but had higher MSE’s. Ridge and PCR performed the worst which is reflected in them having the highest MSE’s