STAT 435 Quiz 2

Pat McCornack

2022-10-10

Q1

a

Simulate the data set. There are 300 observations and 200 variables.

```
set.seed(1)
n = 300; p=200; s=5
x = matrix(rnorm(n * p), n, p)
b = c(rep(1, s),rep(0, p-s))
y = 1 + x %*% b + rnorm(n)
```

b

Create a vector of potential lambda variables.

```
L <- seq(0,2,length.out=100)
```

c

Create a lasso model for the data using the 10th element of the lambda vector L.

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-4
```

```
lasso.model <- glmnet(x, y, lambda = L, alpha = 1)
coef(lasso.model, s = L[10])
```

```
## 201 x 1 sparse Matrix of class "dgCMatrix"
##                    s1
## (Intercept) 0.9769270
## V1          0.7594523
## V2          0.8554086
## V3          0.8465596
## V4          0.8669551
## V5          0.7791742
## V6          .
## V7          .
## V8          .
## V9          .
## V10         .
## V11         .
```

```
## V12          .
## V13          .
## V14          .
## V15          .
## V16          .
## V17          .
## V18          .
## V19          .
## V20          .
## V21          .
## V22          .
## V23          .
## V24          .
## V25          .
## V26          .
## V27          .
## V28          .
## V29          .
## V30          .
## V31          .
## V32          .
## V33          .
## V34          .
## V35          .
## V36          .
## V37          .
## V38          .
## V39          .
## V40          .
## V41          .
## V42          .
## V43          .
## V44          .
## V45          .
## V46          .
## V47          .
## V48          .
## V49          .
## V50          .
## V51          .
## V52          .
## V53          .
## V54          .
## V55          .
## V56          .
## V57          .
## V58          .
## V59          .
## V60          .
## V61          .
## V62          .
## V63          .
## V64          .
## V65          .
```

```
## V66          .
## V67          .
## V68          .
## V69          .
## V70          .
## V71          .
## V72          .
## V73          .
## V74          .
## V75          .
## V76          .
## V77          .
## V78          .
## V79          .
## V80          .
## V81          .
## V82          .
## V83          .
## V84          .
## V85          .
## V86          .
## V87          .
## V88          .
## V89          .
## V90          .
## V91          .
## V92          .
## V93          .
## V94          .
## V95          .
## V96          .
## V97          .
## V98          .
## V99          .
## V100         .
## V101         .
## V102         .
## V103         .
## V104         .
## V105         .
## V106         .
## V107         .
## V108         .
## V109         .
## V110         .
## V111         .
## V112         .
## V113         .
## V114         .
## V115         .
## V116         .
## V117         .
## V118         .
## V119         .
```

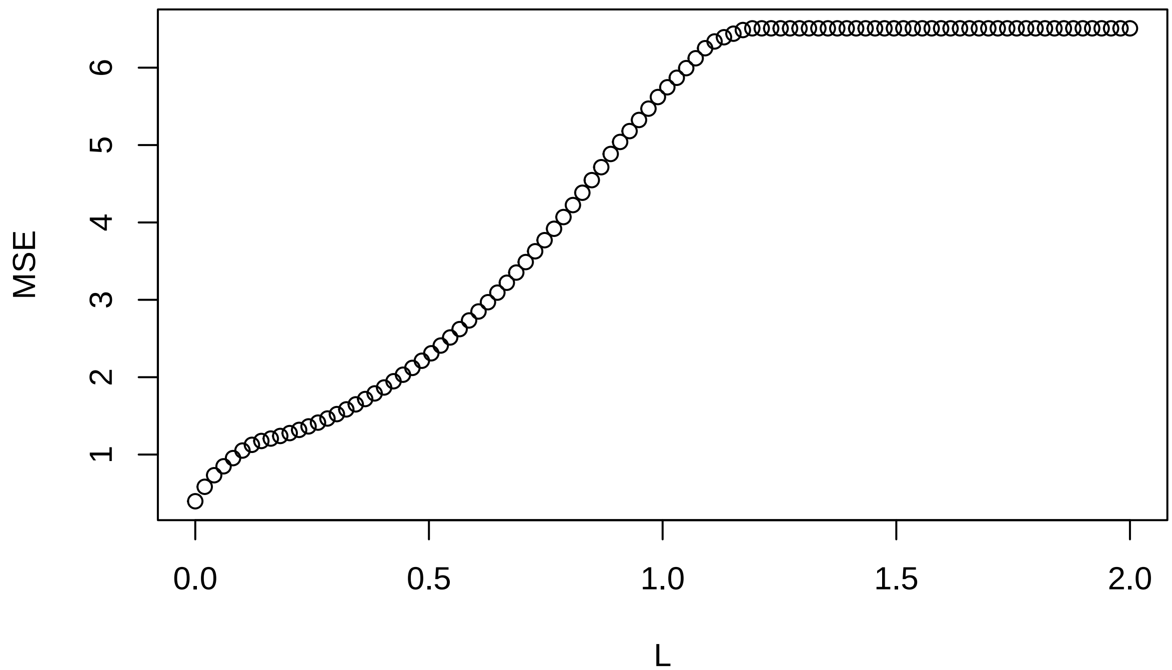
```
## V120          .
## V121          .
## V122          .
## V123          .
## V124          .
## V125          .
## V126          .
## V127          .
## V128          .
## V129          .
## V130          .
## V131          .
## V132          .
## V133          .
## V134          .
## V135          .
## V136          .
## V137          .
## V138          .
## V139          .
## V140          .
## V141          .
## V142          .
## V143          .
## V144          .
## V145          .
## V146          .
## V147          .
## V148          .
## V149          .
## V150          .
## V151          .
## V152          .
## V153          .
## V154          .
## V155          .
## V156          .
## V157          .
## V158          .
## V159          .
## V160          .
## V161          .
## V162          .
## V163          .
## V164          .
## V165          .
## V166          .
## V167          .
## V168          .
## V169          .
## V170          .
## V171          .
## V172          .
## V173          .
```

```
## V174         .
## V175         .
## V176         .
## V177         .
## V178         .
## V179         .
## V180         .
## V181         .
## V182         .
## V183         .
## V184         .
## V185         .
## V186         .
## V187         .
## V188         .
## V189         .
## V190         .
## V191         .
## V192         .
## V193         .
## V194         .
## V195         .
## V196         .
## V197         .
## V198         .
## V199         .
## V200         .
```

d

Compute the mean squared error (MSE) value for each model. The plot shows a monitonic increase in MSE with L until a plateau in MSE at around lambda = 1.2. This suggests a lower lambda value will minimize the MSE.

```r
MSE = c()
for(i in 1:100){
  y.hat <- as.matrix(cbind(1, x)) %*% coef(lasso.model, s = L[i])
  MSE[i] <- mean((y - y.hat )^2)
}

plot(L, MSE)
```



e

The following computes the cross validation error for each value of lambda.

```
k = 5
ncv = ceiling(n/k)  # Observations per fold
cv.ind = rep(1:k, ncv)  # Fold index
cv.ind.random = sample(cv.ind, n, replace = F)  # Randomize fold index
data = data.frame(y = y, x = x)


cv.error = c(); MSE.cv = c()
for(i in 1:100){  # Loop through values of lambda
  for(j in 1:k){  # Loop through folds
      train <- data[cv.ind.random != j, ]
      train.y <- train$y
      lasso.model <- glmnet(train[-1], train.y, lambda = L[i], alpha = 1)

      test = data[cv.ind.random == j,]
      test.values = test$y
      test.response <- as.matrix(cbind(1, test[-1])) %*% coef(lasso.model, s = L[i])
      MSE.cv[j] = mean((test.values - test.response)^2)
    }
  cv.error[i] = mean(MSE.cv)
}
```

```
which.min(cv.error)
```

```
## [1] 7
```

```
L[which.min(cv.error)]
```

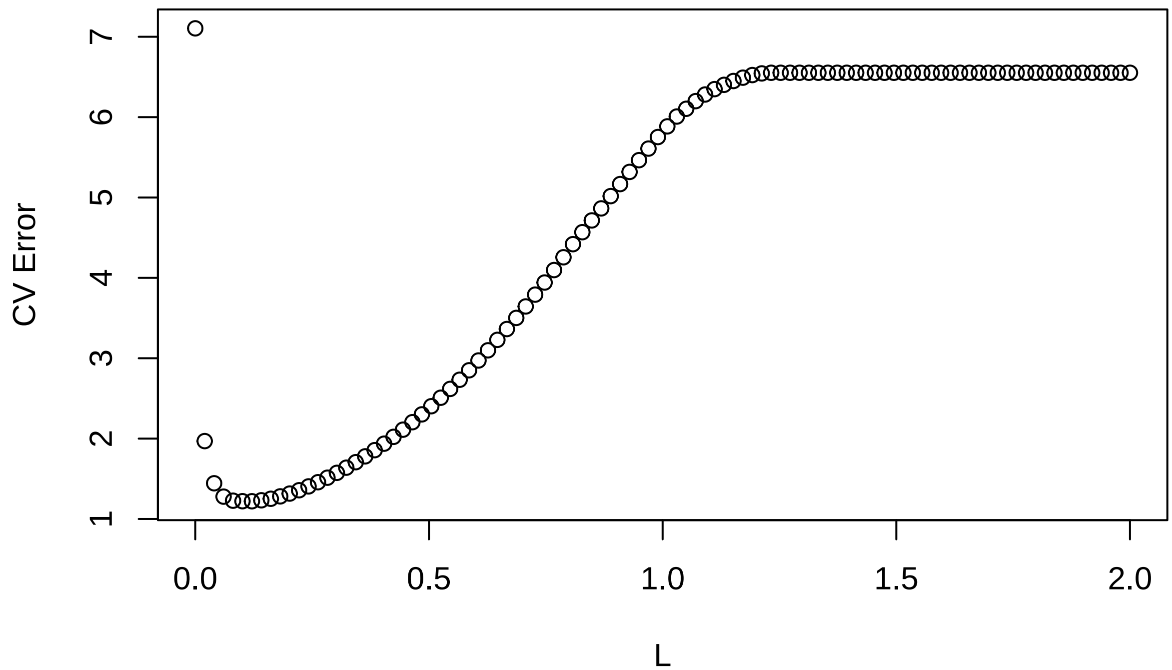
```
## [1] 0.1212121
```

f

```
lasso.funct <- function(x, y, k, L)
{
  ncv = ceiling(dim(x)[1]/k)
  cv.ind = rep(1:k, ncv)
  cv.ind.random = sample(cv.ind, dim(x)[1], replace = F)
  data = data.frame(y = y, x = x)

  cv.error = c(); MSE.cv = c()
  for(i in 1:length(L)){
    for(j in 1:k){
      train <- data[cv.ind.random != j, ]
      train.y <- train$y
      lasso.model <- glmnet(train[-1], train.y, lambda = L[i], alpha = 1)

      test = data[cv.ind.random == j,]
      test.values = test$y
      test.response <- as.matrix(cbind(1, test[-1])) %*% coef(lasso.model, s = L[i])
      MSE.cv[j] = mean((test.values - test.response)^2)
    }
  cv.error[i] = mean(MSE.cv)
  }

  results <- list(coef(lasso.model, s = L[which.min(cv.error)]), cv.error, L, L[which.min(cv.error)])

  return(results)
}
```

g

```
output = lasso.funct(x, y, 5, L)

plot(output[[3]], output[[2]], xlab = 'L', ylab = 'CV Error' )
```


CV Error
1
2
3
4
5
6
7
0.0
0.5
1.0
1.5
2.0
L