STAT409 Homework3

2021320322 / Minseo Yoon

April 11, 2023

1. Assuming floating number representation with B = 10 and d = 4, show that X^TX is not invertible (Hint: Compute its determinant.)

$$X = \begin{bmatrix} 1 & 1.000 \\ 1 & 1.000 \\ 1 & 1.001 \\ 1 & 1.001 \end{bmatrix}$$

$$X^TX = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1.000 & 1.000 & 1.001 & 1.001 \end{bmatrix} \begin{bmatrix} 1 & 1.000 \\ 1 & 1.000 \\ 1 & 1.001 \\ 1 & 1.001 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 4.002 \\ 4.002 & 4.004002 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 4.002 \\ 4.002 & 4.004 \end{bmatrix}$$

Cancellation $4.004002 = (+, 1, .4004)$

$$\det(X^TX) = 4 \times 4.004 - 4.002 \times 4.002$$

$$= 16.016 - 16.016004$$

Cancellation

$$\rightarrow 16.02 - 16.02$$

$16.016 = (+, 2, .1602)$

$16.016004 = (+, 2, .1602)$

$$= 0$$

Since $\det(X^TX) = 0$, X^TX is not invertible.

2. Hitters's data that record salary of Major League Baseball (MLB) players is available in R. You can download the data by running the following code.

```
library(ISLR)
data("Hitters")
Hitters <- na.omit(Hitters) # remove missing values

# response: Salary of Baseball Player
y <- Hitters$Salary # y
# predictors: Players Stats
X <- cbind(Hitters$AtBat, # x1: number of times at batting in the game
           Hitters$Hits, # x2: number of hits
           Hitters$HmRun, # x3: number of homeruns
           Hitters$Runs, # x4: number of runs
           Hitters$Walks, # x5: number of walks
           Hitters$Years) # x6: number of years played in the league
```

```
n <- nrow(X); p <- ncol(X)
my <- mean(y); mX <- apply(X, 2, mean)
y.c <- y - my; X.c <- t(t(X) - mX)
```

(a) Please write your own code to compute the OLS estimator of β, fitted values, and residuals for the linear regression for the Hitters data problem using "qr()" function in "R". (Hint: You may simply apply the code given in the Lecture note)

```
obj <- qr(X.c)
Q <- qr.Q(obj, complete = T)
R <- matrix(0, 6, 6)
R[upper.tri(R, diag = T)] <- obj$qr[upper.tri(obj$qr, diag = T)]

z <- crossprod(Q, y.c)
z1 <- z[1:6]
z2 <- z[-c(1:6)]

beta.hat <- backsolve(R, z1)
beta0.hat <- my - beta.hat %*% mX
c(beta0.hat, beta.hat)
```

```
## [1] -104.106904   -1.963768    9.216153    5.927588   -1.891489    5.838047
## [7]   31.944660
```

```
y.hat <- Q %*% c(z1, rep(0, length(z2))) + my
resid <- Q %*% c(rep(0, length(z1)), z2)
head(cbind(y.hat, resid, y - y.hat))
```

```
##          [,1]       [,2]       [,3]
## [1,] 694.8210 -219.82099 -219.82099
## [2,] 674.7319 -194.73194 -194.73194
## [3,] 784.3457 -284.34567 -284.34567
## [4,] 291.8674 -200.36739 -200.36739
## [5,] 726.4078   23.59224   23.59224
## [6,]  22.5053   47.49470   47.49470
```

(b) Please compare your results obtained from (a) to the result from "lm()" given below:

```
obj <- lm(y ~ X)
est <- coef(obj)
y.hat <- fitted(obj)
resid2 <- resid(obj)
```

```
cbind(c(beta0.hat, beta.hat), est)
```

```
##                                est
## (Intercept) -104.106904 -104.106904
## X1            -1.963768   -1.963768
## X2             9.216153    9.216153
## X3             5.927588    5.927588
## X4            -1.891489   -1.891489
## X5             5.838047    5.838047
## X6            31.944660   31.944660
```

직접 코딩한 QR 분해 기반으로 추정한 β의 값과 lm() 함수에서 추정한 β의 값이 일치한다.

```
head(cbind(resid, resid2))
```

```
##                  resid2
## 1 -219.82099 -219.82099
## 2 -194.73194 -194.73194
## 3 -284.34567 -284.34567
## 4 -200.36739 -200.36739
## 5   23.59224   23.59224
## 6   47.49470   47.49470
```

직접 코딩한 QR 분해 기반으로 계산한 잔차와 lm() 함수에서 계산한 잔차가 일치한다.

3. (Hitters's data) Apply OLS regression, LASSO, Ridge, and Elastic Net Regression to Hitters's data (with appropriately selected λ for the latter regularized methods), and provide their coefficient estimates.

```r
library(glmnet)

obj.ols <- lm(y ~ X)

obj <- glmnet(X, y)
grid <- obj$lambda

obj.cv.rr <- cv.glmnet(X, y, alpha = 0, lambda = grid)
obj.cv.lasso <- cv.glmnet(X, y, alpha = 1, lambda = grid)
obj.cv.net <- cv.glmnet(X, y, alpha = 0.5, lambda = grid)

obj.rr <- glmnet(X, y, alpha = 0, lambda = obj.cv.rr$lambda.min)
obj.lasso <- glmnet(X, y, alpha = 1, lambda = obj.cv.lasso$lambda.min)
obj.net <- glmnet(X, y, alpha = 0.5, lambda = obj.cv.net$lambda.min)
```

```r
result <- cbind(coef(obj.ols), coef(obj.rr), coef(obj.lasso), coef(obj.net))
colnames(result) <- c("OLS", "Ridge", "LASSO", "ElasticNet")
result
```

```
## 7 x 4 sparse Matrix of class "dgCMatrix"
##                     OLS       Ridge       LASSO  ElasticNet
## (Intercept) -104.106904 -115.303221 -106.950234 -116.284234
## V1            -1.963768   -1.709943   -1.899884   -1.675711
## V2             9.216153    8.259815    8.957733    8.048560
## V3             5.927588    5.585022    5.769301    5.293466
## V4            -1.891489   -1.449306   -1.709945   -1.115634
## V5             5.838047    5.601170    5.748729    5.465046
## V6            31.944660   32.132677   32.012369   32.223001
```