Lab 4

Jaime Lin

11:59PM March 10, 2021

Load up the famous iris dataset. We are going to do a different prediction problem. Imagine the only input x is Species and you are trying to predict y which is Petal.Length. A reasonable prediction is the average petal length within each Species. Prove that this is the OLS model by fitting an appropriate `lm` and then using the predict function to verify.

```
data(iris)
mod = lm(Petal.Length~Species, iris)

mean(iris$Petal.Length[iris$Species == "setosa"])

## [1] 1.462

mean(iris$Petal.Length[iris$Species == "versicolor"])

## [1] 4.26

mean(iris$Petal.Length[iris$Species == "virginica"])

## [1] 5.552

predict(mod, data.frame(Species = c("setosa")))

##     1
## 1.462

predict(mod, data.frame(Species = c("versicolor")))

##    1
## 4.26

predict(mod, data.frame(Species = c("virginica")))

##     1
## 5.552
```

Construct the design matrix with an intercept, X, without using `model.matrix`.

```
X <- cbind(1, iris$Species == "versicolor",iris$Species == "virginica")
head(X)

##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    1    0    0
## [3,]    1    0    0
## [4,]    1    0    0
```

```
## [5,]    1    0    0
## [6,]    1    0    0
```

Find the hat matrix H for this regression.

```
H = X %*% solve(t(X) %*% X) %*% t(X)
Matrix::rankMatrix(H)
```

```
## [1] 3
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 3.330669e-14
```

Verify this hat matrix is symmetric using the `expect_equal` function in the package `testthat`.

```
pacman::p_load(testthat)
expect_equal(H, t(H))
```

Verify this hat matrix is idempotent using the `expect_equal` function in the package `testthat`.

```
expect_equal(H, H%*% H)
```

Using the `diag` function, find the trace of the hat matrix.

```
sum(diag(H))
```

```
## [1] 3
```

It turns out the trace of a hat matrix is the same as its rank! But we don't have time to prove these interesting and useful facts..

For masters students: create a matrix $X_\perp$.

```
#TO-DO
```

Using the hat matrix, compute the $\hat{y}$ vector and using the projection onto the residual space, compute the e vector and verify they are orthogonal to each other.

```
y = iris$Petal.Length
y_hat = H %*% y
e = (diag(nrow(iris))-H) %*% y
```

Compute SST, SSR and SSE and R^2 and then show that SST = SSR + SSE.

```
#TO-DO
SSE = t(e) %*% e
y_bar = mean(y)
SST = t(y - y_bar) %*% (y-y_bar)
```

```
Rsq = 1 - SSE/SST
SSR = t(y_hat - y_bar) %*% (y_hat - y_bar)
SSR

##          [,1]
## [1,] 437.1028

expect_equal(SSR+SSE, SST)
```

Find the angle θ between y - $\bar{y}1$ and $\hat{y} - \bar{y}1$ and then verify that its cosine squared is the same as the R^2 from the previous problem.

```
#TO-DO
theta = acos(t(y - y_bar) %*% (y_hat-y_bar) /sqrt(SST * SSR))
theta * (180/pi)

##          [,1]
## [1,] 14.01245
```

Project the y vector onto each column of the X matrix and test if the sum of these projections is the same as yhat.

```
proj_1 = (X[,1] %*% t(X[,1])/ as.numeric(t(X[,1]) %*% X[,1])) %*% y
proj_2 =(X[,2] %*% t(X[,2])/ as.numeric(t(X[,2]) %*% X[,2]) )%*% y
proj_3 =(X[,3] %*% t(X[,3])/ as.numeric(t(X[,3]) %*% X[,3])) %*% y

#expect_equal(proj_1+proj_2+proj_3, y_hat)
# The sum of projections is not equal to y_hat
```

Construct the design matrix without an intercept, X, without using `model.matrix`.

```
#TO-DO
V = cbind(proj_1,proj_2, proj_3)
```

Find the OLS estimates using this design matrix. It should be the sample averages of the petal lengths within species.

```
V_1 = as.numeric(((t(X[,1]) %*% X[,1])^(-1)) %*% t(X[,1]) %*% y_hat)
```

Verify the hat matrix constructed from this design matrix is the same as the hat matrix constructed from the design matrix with the intercept. (Fact: orthogonal projection matrices are unique).

```
V_2 = mean(y_hat)
expect_equal(V_1,V_2)
```

Project the y vector onto each column of the X matrix and test if the sum of these projections is the same as yhat.

```
z_1 = sum(y_hat)
z_2 = as.numeric(X[,1] %*% y_hat)
expect_equal(z_1, z_2)
```

Convert this design matrix into Q, an orthonormal matrix.

```
Q = X[,1] %*% t(X[,1])
```

Project the y vector onto each column of the Q matrix and test if the sum of these projections is the same as yhat.

```
proj_4 = (Q[,1] %*% t(Q[,1])/ as.numeric(t(Q[,1]) %*% Q[,1])) %*% y
#expect_equal(proj_4, y_hat)
#The sum of projections is not equal to y_hat
```

Find the $p = 3$ linear OLS estimates if Q is used as the design matrix using the `lm` method. Is the OLS solution the same as the OLS solution for X?

```
V_3 = ((t(X[,3]) %*% X[,3])^(-1)) %*% t(X[,3]) %*% Q

#expect_equal(V_1, V_3)
#They are not equal the length are not the same
```

Use the predict function and ensure that the predicted values are the same for both linear models: the one created with X as its design matrix and the one created with Q as its design matrix.

```
#TO-DO
```

Clear the workspace and load the boston housing data and extract X and y. The dimensions are $n = 506$ and $p = 13$. Create a matrix that is $(p + 1) \times (p + 1)$ full of NA's. Label the columns the same columns as X. Do not label the rows. For the first row, find the OLS estimate of the y regressed on the first column only and put that in the first entry. For the second row, find the OLS estimates of the y regressed on the first and second columns of X only and put them in the first and second entries. For the third row, find the OLS estimates of the y regressed on the first, second and third columns of X only and put them in the first, second and third entries, etc. For the last row, fill it with the full OLS estimates.

```
y = MASS::Boston[, 14]
X = as.matrix(cbind(1, MASS::Boston[, 1 : 13]))
B = matrix(NA, 14, 14)
H = X %*% solve(t(X) %*% X) %*% t(X)
y_hat = H %*% y
((t(X[,1]) %*% X[,1])^(-1)) %*% t(X[,1]) %*% y_hat

##          [,1]
## [1,] 22.53281

((t(X[,1:2]) %*% X[,1:2])^(-1)) %*% t(X[,1:2]) %*% y_hat

##          [,1]
## 1    36.58143
## crim  6.81988

((t(X[,1:3]) %*% X[,1:3])^(-1)) %*% t(X[,1:3]) %*% y_hat
```

```
##            [,1]
## 1      65.90452
## crim  366.55285
## zn     57.28344
```

Why are the estimates changing from row to row as you add in more predictors?

Because each column that represent a variable have different sum. When you add two columns from the matrix the sum of the first and the sum of the second is going to get a new sum.

Create a vector of length $p + 1$ and compute the R^2 values for each of the above models.

```
#TO-DO
```

Is R^2 monotonically increasing? Why?

#TO-DO