Homework 4

Greg DePaul

2023-03-28

Problem 6 - Multiple regression by matrix algebra in R.

You need to submit your codes alongside the answers, plots, outputs, etc. You are required to use R Markdown: Please submit a .rmd file and its corresponding .html file.

Consider the following data set with 5 cases, one response variable Y and two predictor variables X_1, X_2.

case	Y	X1	X2
1	-0.97	-0.63	-0.82
2	2.51	0.18	0.49
3	-0.19	-0.84	0.74
4	6.53	1.60	0.58
5	1.00	0.33	-0.31

Consider the first-order model for the following questions:

```
Y <- c(-0.97, 2.51, -0.19, 6.53, 1.00)
X_1 <- c(-0.63, 0.18, -0.84, 1.60, 0.33)
X_2 <- c(-0.82, 0.49, 0.74, 0.58, -0.31)
ones <- rep(1, length(Y))
X <- matrix(c(ones, X_1, X_2), ncol = 3)
print(X)
```

(a) Create the design matrix X and the response vector Y. Calculate $X'X$, $X'Y$ and $(X'X)^{-1}$.

```
##      [,1]  [,2]  [,3]
## [1,]    1 -0.63 -0.82
## [2,]    1  0.18  0.49
## [3,]    1 -0.84  0.74
## [4,]    1  1.60  0.58
## [5,]    1  0.33 -0.31
```

```
print(Y)
```

```
## [1] -0.97  2.51 -0.19  6.53  1.00
```

```
print(t(X) %*% X)
```

```
##      [,1]   [,2]   [,3]
## [1,] 5.00 0.6400 0.6800
## [2,] 0.64 3.8038 0.8089
## [3,] 0.68 0.8089 1.8926
```

```r
print(t(X) %*% Y)
```

```
##         [,1]
## [1,]  8.8800
## [2,] 12.0005
## [3,]  5.3621
```

```r
print( solve( (t(X) %*% X) ) )
```

```
##             [,1]        [,2]        [,3]
## [1,]  0.21184719 -0.02140278 -0.06696786
## [2,] -0.02140278  0.29134054 -0.11682948
## [3,] -0.06696786 -0.11682948  0.60236791
```

```r
beta <- solve(t(X) %*% X) %*% (t(X) %*% Y)
print(beta)
```

(b) Obtain the least-squares estimators $\hat{\beta}$.

```
##          [,1]
## [1,] 1.265271
## [2,] 2.679724
## [3,] 1.233270
```

```r
H <- X %*% solve(t(X) %*% X) %*% t(X)
print(H)
```

(c) Obtain the hat matrix H. What are $rank(H)$ and $rank(I - H)$?

```
##             [,1]       [,2]        [,3]       [,4]        [,5]
## [1,]  0.74859901 0.02181768  0.01132102 -0.1770289  0.39529119
## [2,]  0.02181768 0.27197293  0.35049579  0.2534024  0.10231125
## [3,]  0.01132102 0.35049579  0.82936038 -0.1072487 -0.08392853
## [4,] -0.17702890 0.25340235 -0.10724866  0.7973084  0.23356681
## [5,]  0.39529119 0.10231125 -0.08392853  0.2335668  0.35275928
```

```r
print(rankMatrix(H))
```

```
## [1] 3
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 1.110223e-15
```

```r
print(rankMatrix(diag(length(Y)) - H))
```

```
## [1] 2
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 1.110223e-15
```

```
print(tr(H))
```

(d) Calculate the trace of H and compare it with $rank(H)$ from part (c). What do you find?

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

This makes sense because

$$tr(H) = rank(X) = p = 3 = rank(H)$$

```
print(X %*% beta)
```

(e) Obtain the fitted values, the residuals, SSE and MSE. What should be the degrees of freedom of SSE?

```
##             [,1]
## [1,] -1.43423719
## [2,]  2.35192330
## [3,] -0.07307774
## [4,]  6.26812586
## [5,]  1.76726576
```

```
print(Y - X %*% beta)
```

```
##            [,1]
## [1,]  0.4642372
## [2,]  0.1580767
## [3,] -0.1169223
## [4,]  0.2618741
## [5,] -0.7672658
```

```
SSE <- sum( ( Y - X %*% beta )^2 )
print(SSE)
```

```
## [1] 0.91145
```

```
print(SSE / (length(Y) - 3))
```

```
## [1] 0.455725
```

We expect

$$df(SSE) = n - p = 5 - 3 = 2$$

Consider the nonadditive model with interaction between X1 and X2 for the following questions:

```
X_1X_2 = X_1 * X_2
X <- matrix(c(ones, X_1, X_2, X_1X_2), ncol = 4)
H <- X %*% solve(t(X) %*% X) %*% t(X)
print(rankMatrix(H))
```

(f) Create the design matrix. Obtain the hat matrix H. Find $rank(H)$ and $rank(I-H)$. Compare the ranks with those from part (c), what do you observe?

```
## [1] 4
## attr(,"method")
```

```
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 1.110223e-15
```

```r
print(rankMatrix(diag(length(Y)) - H))
```

```
## [1] 2
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 1.110223e-15
```

```r
beta <- solve(t(X) %*% X) %*% (t(X) %*% Y)
print(beta)
```

(g) Obtain the least-squares estimators $\hat{\beta}$.

```
##          [,1]
## [1,] 1.051738
## [2,] 1.987286
## [3,] 1.804233
## [4,] 1.387774
```

```r
print(X %*% beta)
```

(h) Obtain the fitted values, the residuals, SSE and MSE. What should be the degrees of freedom of SSE?

```
##            [,1]
## [1,] -0.9627998
## [2,]  2.4159250
## [3,] -0.1450905
## [4,]  6.5657047
## [5,]  1.0062607
```

```r
print(Y - X %*% beta)
```

```
##              [,1]
## [1,] -0.007200196
## [2,]  0.094075045
## [3,] -0.044909459
## [4,] -0.035704724
## [5,] -0.006260666
```

```r
SSE <- sum( ( Y - X %*% beta )^2 )
print(SSE)
```

```
## [1] 0.01223284
```

```r
print(SSE / (length(Y) - 4))
```

```
## [1] 0.01223284
```

(i) Which of the two models appears to fit the data better? The second order non-additive model appears to fit the data better based off of the decrease in SSE. But this might be because we are overfitting the data because the model complexity exceeds the number of data points available.