Lecture 9: Least Squares Estimation using Matrices
STAT 632, Spring 2020

Mathematical Preliminaries

Let

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{pmatrix}$$

be an $n \times 1$ vector, and $f(\boldsymbol{\theta})$ a scalar function of $\boldsymbol{\theta}$. The derivative of $f(\boldsymbol{\theta})$ with respect to the vector $\boldsymbol{\theta}$ is defined as

$$\frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{pmatrix} \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_1} \\ \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_2} \\ \vdots \\ \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_n} \end{pmatrix}$$

Recall from calculus two important rules of differentiation for a scalar a and variable x:

1. $\frac{d}{dx} ax = a$
2. $\frac{d}{dx} ax^2 = 2ax$

There are similar rules when taking a derivative with respect to a vector:

1. Let $\boldsymbol{c'} = \begin{pmatrix} c_1 & c_2 & \cdots & c_n \end{pmatrix}$ and $f(\boldsymbol{\theta}) = \boldsymbol{c'\theta}$, then it follows that

$$\frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \boldsymbol{c}$$

2. Let $\boldsymbol{A}$ be an $n \times n$ symmetric matrix and $f(\boldsymbol{\theta}) = \boldsymbol{\theta' A \theta}$, then it follows that

$$\frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 2\boldsymbol{A\theta}$$

Your turn: Prove the first rule for vector differentiation.

Recall the matrix notation for multiple linear regression (MLR):

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{e}$$

Your turn: What are the dimensions of each term?

For MLR, the residual sum of squares, as a function of the vector $\boldsymbol{\beta}$, can be written in matrix notation as

$$R(\boldsymbol{\beta}) = (\boldsymbol{Y} - \boldsymbol{X\beta})'(\boldsymbol{Y} - \boldsymbol{X\beta})$$

Expanding this equation out gives:

$$\begin{aligned} R(\boldsymbol{\beta}) &= \boldsymbol{Y'Y} - \boldsymbol{Y'X\beta} - (\boldsymbol{X\beta})'\boldsymbol{Y} + (\boldsymbol{X\beta})'(\boldsymbol{X\beta}) \\ &= \boldsymbol{Y'Y} - \boldsymbol{Y'X\beta} - \boldsymbol{\beta'X'Y} + \boldsymbol{\beta'X'X\beta} \\ &= \boldsymbol{Y'Y} - 2\boldsymbol{Y'X\beta} + \boldsymbol{\beta'X'X\beta} \end{aligned}$$

Your turn: Argue that $\boldsymbol{Y'X\beta} = \boldsymbol{\beta'X'Y}$, so the terms can be combined.

Next, to find the least squares estimates, we minimize $R(\boldsymbol{\beta})$ by taking the derivative, with respect to the vector $\boldsymbol{\beta}$, and setting the derivative equal to zero. Since $\boldsymbol{X'X}$ is a symmetric matrix, we can use the two rules for vector differentiation to accomplish this.

$$\frac{\partial R(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \boldsymbol{0} - 2\boldsymbol{X'Y} + 2\boldsymbol{X'X\beta}$$

Setting the derivative equal to zero gives the **normal equations** for MLR:

$$\boldsymbol{X'X\beta} = \boldsymbol{X'Y}$$

Assuming that $\boldsymbol{X'X}$ is invertible, the vector of least squares estimates is given by

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'Y}$$

Note that $\boldsymbol{X}'\boldsymbol{X}$ is not invertible (singular) when the columns of $\boldsymbol{X}$ are **linearly dependent**. That is, an explanatory variable can be expressed as a linear combination of other explanatory variables in the model. For observational data, this is usually caused by some oversight. Here are some examples:

- A person's weight is measured in both pounds and kilos and both explanatory variables are included in the model.
- For each student we record their verbal SAT score, math SAT score, and total SAT score, and then include all three variables in the model.

Using the menu pricing data set (lecture 7), the code below demonstrates how to manually compute the vector $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$ of least squares estimates.

```
nyc <- read.csv("https://ericwfox.github.io/data/nyc.csv")

# response vector
Y <- matrix(nyc$Price, ncol=1)

# design matrix
X <- cbind(Intercept = 1, nyc[,c("Food", "Decor", "East")])
X <- as.matrix(X)
rownames(X) <- nyc$Restaurant
X[1:4,]

##                     Intercept Food Decor East
## Daniella Ristorante         1   22    18    0
## Tello's Ristorante          1   20    19    0
## Biricchino                  1   21    13    0
## Bottino                     1   20    20    0

# manually calculate least squares estimates
betaHat <- solve(t(X) %*% X) %*% t(X) %*% Y
betaHat

##                 [,1]
## Intercept -24.026880
## Food        1.536346
## Decor       1.909373
## East        2.067013

# compare with lm()
lm1 <- lm(Price ~ Food + Decor + East, data=nyc)
coef(lm1)

## (Intercept)        Food       Decor        East
##  -24.026880    1.536346    1.909373    2.067013
```