MA478 - Lesson2

Clark

A couple of things to refresh prior to today's lesson. A matrix $X_{n\times n}$ is invertible if:

The column space (or the span) of X is the set of all possible linear combinations of its column vectors. For instance:

The last thing I want to talk about, and we won't go too much into this, but I think it's helpful to see what's going on in a regressio model, is a *projection matrix*. In broad terms, a projection matrix is a square matrix $(n \times n)$ that projects a point in $\mathbb{R}_n$ to a subspace. I think the simplest example is:

```
Px = matrix(c(.5,.5,.5,.5),nrow=2,ncol=2)

print(Px)
```

```
##      [,1] [,2]
## [1,]  0.5  0.5
## [2,]  0.5  0.5
```

```
y = matrix(c(5,3),nrow=2,ncol=1)

Px %*% y
```

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

```
iden = matrix(c(1,0,0,1),ncol=2,nrow=2)

(iden - Px) %*% y
```

```
##      [,1]
## [1,]    1
## [2,]   -1
```

```
Px%*%Px%*%y
```

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

```
Px %*% (iden-Px)%*%y
```

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

```
Px %*% y + (iden-Px)%*%y
```

```
##      [,1]
## [1,]    5
## [2,]    3
```

Here we see a couple of properties of projection matrices. They are *idempotent*, they are symmetric, the identity minus the projection matrix is also a projection matrix and projects to the orthogonal space.

So, let's see why all of this is relevant to our discussion of linear regression.

Let's start by writing out the linear regression model in matrix form:

There's a couple of assumptions we typically make.

If we add an additional normality assumption we can write the entire multiple regression model as:

To fit this model, we want to find $\hat{\beta}$ that minimizes the squared error between y and $\hat{y} = x\beta$. Or in other words we want to minimize the expression:

While we could do this with summations, it turns out this is a pain for any model beyond the simple linear regression model. Here linear algebra really helps us.

We can write out the expression we are trying to minimize as a function of β

Therefore, to find the β that minimizes the expression, perhaps we take the partial with respect to β and set it equal to zero.

If we use this $\hat{\beta}$ it turns out that $X\hat{\beta}$ projects y onto the column space of X.

That is:

The big take away here is, if we use the OLS estimate of β, which is the same as the MLE of β if we assume ϵ has a multi-variate normal distribution, we end up with a $\hat{y}$ that is the projection of y onto the column space of X, which we sometimes call the design space or the model space.

So, that's it. To fit a linear regression model we have a pretty simple algorithm. We construct X, we compute $(X^TX)^{-1}$, we return $\hat{\beta}$

```
library(faraway)
library(tidyverse)
library(reshape2)

data(mtcars)

null_lm = lm(mpg ~ 1,data=mtcars)

x = model.matrix(null_lm)

dim(x)
```

```
## [1] 32  1
```

```r
beta_hat <- function(X,y){
  beta_val = solve(t(X)%*%X)%*%t(X)%*%y
  return(beta_val)
}

Px <- function(X){
  return(X%*%solve(t(X)%*%X)%*%t(X))
}
```

Take a look at the projection matrix. What space are we projecting y onto?

Let's take a look at determine if there's a relationship between Engine shape and miles per gallon.

Build out a regression model and find $\hat{\beta}$ and P_x. Ensure you treat engine types as factors here.

Check your answers with what you get when you run `lm`. What additional output do you get from `lm` that might be of value?

The additional output form `lm` comes from additional assumptions we can make about ϵ. This additional assumption helps to address to questions. IS $\beta_j = 0$ or not? OR are all regression coefficients equal to zero (meaning, is our model any better than the null model).

Let's look at the first of these tests. To test this, we need the distrubtion of $\hat{\beta}$. I'm not going to derive this here, but if we assume ϵ is multivariate normal, $\hat{\beta} \sim MVN(\beta, (X^T X)^{-1}\sigma^2)$.

Typically, we don't know σ^2 though, so instead we form a t statistic using $t = \frac{\hat{\beta}}{se(\hat{\beta})} \sim t_{n-p}$

```
estimate = beta_hat(x,mtcars$mpg)
SSE = sum((mtcars$mpg - x%*%estimate)^2)
p = qr(x)$rank
n = length(mtcars$mpg)
hat_sigma_sq = SSE/(n-p)

se_beta = solve(t(x)%*%x)*hat_sigma_sq

tstat=estimate/sqrt(se_beta)



print(paste0("Estimate - ",formatC(estimate, digits = 2, format ="f"),"  Std. Error - ",
             formatC(sqrt(hat_sigma_sq),digits = 2, format = "f"),
             "   t value - ",format(tstat,digits =5, format="f") ))
```

```
## [1] "Estimate - 20.09  Std. Error - 6.03   t value - 18.857"
```

Which we can compare to

```
summary(null_lm)
```

```
##
## Call:
## lm(formula = mpg ~ 1, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6906 -4.6656 -0.8906  2.7094 13.8094
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.091      1.065   18.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.027 on 31 degrees of freedom
```

Repeat this using Engine Shape. Note that you will have to change the code for se_beta as it's now a matrix instead of a scalar (why?).

The last hypothesis test that might be of value is testing whether the model is preferable to the null model. That is:

To do this, we conduct a likelihood ratio test. Typically this is done by:

```
bigger_lm = lm(mpg ~ vs,data=mtcars)

x_bigger = model.matrix(bigger_lm)

estimate_bigger = beta_hat(x_bigger,mtcars$mpg)
SSE_bigger = sum((mtcars$mpg - x_bigger%*%estimate_bigger)^2)
p_bigger = qr(x_bigger)$rank

F_stat = ((SSE-SSE_bigger)/(p_bigger-p))/(SSE_bigger/(n-p_bigger))

print(paste0("Our F Statistic is - ",
             formatC(F_stat,digits=5,format = "f")))
```

```
## [1] "Our F Statistic is - 23.66224"
```

In R this is found through:

```
anova(null_lm,bigger_lm)
```

Next lesson: we will expand upon linear model inference via a deeper dive of model diagnostics.

Agresti: 2.5, 3.1 (up to 3.1.4), skim 3.2-3.3, read 3.4 and 3.5 Faraway: Pages 14-16 (Diagnostics)