Problem Set 7

Jeffrey Kwarsick

November 17, 2017

1 Problem 1

To determine if the standard error of the estimate of the coefficient properly characterizes the uncertainty of the estimated regression coefficient, I would first calculate the variance of my estimate of the error. This will tell me how uncertain we are about my estimate of the standard error. I would then calculate the variance of that variance of the previously calculated variance as this would give me a measure of the robustness of my estimate of the standard error to the variance. These two quantities together will help determine if the standard error properly characterizes the uncertainity of the estimated regression coefficient.

2 Problem 2

Definition of matrix 2-norm:

$$\|A\|_2 = max_{\|z\|_2=1} = \frac{\|Az\|_2}{\|z\|_2}$$

We also know that:

$$\|Az\|_2^2 = z^T A^T Az$$

Because A is symmetric, A^T is also symmetric and $A^T A$ is symmetric and positive semi-definite, decomposition is possible as the following:

$$A^T A = UDU^T$$

Where D is a diagonal matrix containing the eigenvalues of matrix A in the main diagonal and U and U^T are orthogonal matrices. Substitunting the decomposition in, we get:

$$\frac{\|Az\|_2}{\|z\|_2} = \frac{z^T A^T Az}{z^T z}$$

$$\frac{\|Az\|_2}{\|z\|_2} = \frac{z^T UDU^T z}{z^T UU^T z}$$

We then set $y = U^T z$ and substitute in. y is another unit vector of magnitude 1.

$$\frac{\|Az\|_2}{\|z\|_2} = \frac{y^T Dy}{y^T y}$$

$$\frac{\|Az\|_2}{\|z\|_2} = \frac{\sum_{i=1}^n \sigma_i^2 |y_i|^2}{\sum_{i=1}^n |y_i|^2}$$

$$\frac{\|Az\|_2}{\|z\|_2} \leq \sigma_1^2$$

The inequality must be true for all non-zero z. Because U is orthogonal, there exists a z that satisfies the following:

$$y = U^T z = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = e_1$$

Therefore:

$$z^T A^T A z = e_1 \Sigma e_1 = \sigma_1^2$$

Therefore, we can conclude that:

$$\|A\|_2 = sup_{z:\|z\|_2=1} \sqrt{(Az)^T Az} = \sigma_1(\lambda_{max})$$

To solve this problem I reviewed the Linear Algebra textbook *Linear Algebra with Applications, 8 ed.* by *Steven J. Leon.* I specifically referred to the chapter on numerical linear algebra.

3 Problem 3

3.1 Part (a)

Consider the rectangular matrix X_{nxp} and it's decomposition.

$$X = U_{nxp} \Lambda_{nxp} V_{pxp}^T$$

The transpose of X is as follows:

$$X^T = (U \Lambda V^T)^T$$
$$X^T = V \Lambda^T U^T$$

Λ is the diagonal matrix with the elements along its main diagonal are the eigenvalues of matrix X. Because of this,

$$\Lambda^T = \Lambda$$

and therefore:

$$X^T X = V \Lambda^T U^T U \Lambda V^T$$
$$= V \Lambda^T \Lambda V^T$$
$$X^T X = V \Lambda^2 V^T$$

The matrices V and V^T are the eigenvectors of our matrix X. More specifically, they are teh right vectors of matrix X. S^2 is the matrix containing the square of the eigenvalues of matrix X, or the singular values of X. To show that $X^T X$ is semi-positive definite, take a vector, v, such that all elements of v are non-zero.

$$v^T X^T X v = (Xv)^T X v$$

$$v^T X^T X v = (Xv)^T X v$$
$$v^T X^T X v = \sigma^T \sigma$$
$$v^T X^T X v = \sum_i^n \sigma_i^2 \geq 0$$

```
#define size of n and p such that n > p
n <- 400
p <- 200

#Generate matrix and compute eigenvalues and vectors
#and single value decomposition
X <- matrix(runif(n*p), ncol = p)
eigen.X <- eigen(t(X) %*% X)
X.svd <- svd(X)

####
```

```r
# Demonstrating the right singular vectors of X are equal
# the eigenvectors of X^T * X
####

#Right Singular Vectors of X
print(X.svd$v[1:5,1:10])

##             [,1]        [,2]         [,3]        [,4]        [,5]
## [1,] -0.06813677 -0.08790737  0.084678647 -0.02425367  0.05070523
## [2,] -0.07070830 -0.02469730  0.070506339 -0.08271604 -0.05532664
## [3,] -0.07238782 -0.12025519  0.071822423 -0.01316548 -0.10366130
## [4,] -0.07360482 -0.06138583 -0.022185284  0.01096447  0.05733620
## [5,] -0.07201876  0.15476802 -0.007304002  0.11333721  0.04997963
##              [,6]        [,7]        [,8]         [,9]        [,10]
## [1,] -0.009758333  0.03407803 -0.04137081  0.084948038 -0.012269905
## [2,]  0.065789864  0.04974208 -0.01113884  0.013169213 -0.119369645
## [3,] -0.091940108 -0.05510131 -0.08520849  0.007023418 -0.019401255
## [4,]  0.071919088  0.04574209  0.03125208  0.017988699  0.140272966
## [5,]  0.066719504  0.08224127  0.13259886 -0.076396869 -0.006794985

#Eigenvectors of X^T * X
print(eigen.X$vectors[1:5,1:10])

##             [,1]        [,2]         [,3]        [,4]        [,5]
## [1,] -0.06813677  0.08790737 -0.084678647 -0.02425367  0.05070523
## [2,] -0.07070830  0.02469730 -0.070506339 -0.08271604 -0.05532664
## [3,] -0.07238782  0.12025519 -0.071822423 -0.01316548 -0.10366130
## [4,] -0.07360482  0.06138583  0.022185284  0.01096447  0.05733620
## [5,] -0.07201876 -0.15476802  0.007304002  0.11333721  0.04997963
##              [,6]        [,7]        [,8]         [,9]        [,10]
## [1,] -0.009758333 -0.03407803 -0.04137081  0.084948038  0.012269905
## [2,]  0.065789864 -0.04974208 -0.01113884  0.013169213  0.119369645
## [3,] -0.091940108  0.05510131 -0.08520849  0.007023418  0.019401255
## [4,]  0.071919088 -0.04574209  0.03125208  0.017988699 -0.140272966
## [5,]  0.066719504 -0.08224127  0.13259886 -0.076396869  0.006794985

####
# Demonstrating the singular values of X are equal
# the eigenvalues of X^T * X
####

#Squared Singular Values of X
print(X.svd$d[1:10]**2)

##  [1] 20157.52513    93.52400    91.36626    88.84670    86.84386
##  [6]    85.66742    84.49989    83.61075    82.95099    81.85568

#Eigenvalues of X^T * X
print(eigen.X$values[1:10])

##  [1] 20157.52513    93.52400    91.36626    88.84670    86.84386
##  [6]    85.66742    84.49989    83.61075    82.95099    81.85568

####
# Using a library in R to determine if X^T * X is positive semi-definite
```

```
####

library(matrixcalc)
is.positive.semi.definite(t(X) %*% X)

## [1] TRUE
```

Note: Used *Linear Algebra with Applications, 8 ed.* as a reference to complete this problem.

3.2 Part (b)

Now consider the $n \times n$ matrix Σ with the assumption that the eigendecomposition has been computed for matrix Σ.

$$\Sigma D \Sigma^T + Ic = \Sigma D \Sigma^T + \Sigma c \Sigma^T$$

$$Z = \Sigma(D + cI)\Sigma^T$$

An example block of code to demonstrate above equations at play.

```
N <- 50
SIG <- crossprod(matrix(rnorm(N^2), N))
eigen.SIG <- eigen(SIG)

#Create Z
Z <- SIG + diag(N)*rep(2,50)
eigen.Z <- eigen(Z)

#"Control" for comparison, slow
eigen.Z$values

##  [1] 208.454887 176.749233 153.068039 147.096495 139.008874 136.051445
##  [7] 127.550437 121.630441 107.549925  97.534315  93.775923  89.768436
## [13]  88.462158  77.255819  73.932010  66.240308  65.812788  54.765629
## [19]  53.727563  49.707347  46.792962  41.582659  41.286670  39.543724
## [25]  38.799382  34.295566  28.896615  25.679228  23.876621  22.619509
## [31]  19.862212  17.309528  16.702400  15.211780  11.762190  10.891793
## [37]   9.609792   8.714730   7.269563   6.560287   6.312309   5.063596
## [43]   4.127297   3.556234   3.211390   2.689986   2.450144   2.259781
## [49]   2.113140   2.060030

#O(n) calculation, fast
eigen.SIG$values + rep(2,50)

##  [1] 208.454887 176.749233 153.068039 147.096495 139.008874 136.051445
##  [7] 127.550437 121.630441 107.549925  97.534315  93.775923  89.768436
## [13]  88.462158  77.255819  73.932010  66.240308  65.812788  54.765629
## [19]  53.727563  49.707347  46.792962  41.582659  41.286670  39.543724
## [25]  38.799382  34.295566  28.896615  25.679228  23.876621  22.619509
## [31]  19.862212  17.309528  16.702400  15.211780  11.762190  10.891793
## [37]   9.609792   8.714730   7.269563   6.560287   6.312309   5.063596
## [43]   4.127297   3.556234   3.211390   2.689986   2.450144   2.259781
## [49]   2.113140   2.060030
```

4 Problem 4

4.1 Part (a)

To start, let us consider the equation provided in this problem.

$$\hat{\beta} = C^{-1}d + C^{-1}A^T(AC^{-}1A^T)^{-1}(-AC^{-1}d + b)$$

First, I would compute the $C^{-1}A^T(AC^{-}1A^T)^{-1}$ resulting in a $p \times m$ matrix. Then I would compute the resulting matrix of $C^{-1}d$, which would be of the size $p \times 1$. Then I would compute the result of $AC^{-1}d$ which would be of size $m \times 1$. Then, I would compute $C^{-1}d$ which would result in a $p \times 1$ matrix.

With all of the parts computed, I would then move forward to compute $\hat{\beta}$. First I would deal with the follwing component of the equation:

$$C^{-1}A^T(AC^{-}1A^T)^{-1}(-AC^{-1}d + b)$$

I would then add the resulting $p \times 1$ matrices with the result of $C^{-1}d$ (also a $p \times 1$ matrix). Overall, this would give me $\hat{\beta}$, a $p \times 1$ matrix.

4.2 Part (b)

Did not get to this part of the problem set.

5 Problem 5

5.1 Part (a)

You cannot calculate the complete the calculation in the two stages as given by the following:

$$\hat{X} = Z(Z^TZ)^{-1}Z^TX$$

$$\hat{\beta} = (\hat{X}^T\hat{X})^{-1}\hat{X}^Ty$$

This is not possible even if you use OLS techniques for each stage because of the sizes of the matrices involved. Matriz Z, is 60 million rows by 630 columns and despite the matrix being sparse, it would still require a significant amount of memory in order to implement. The sheer size of these matrices would also drastically affect the computation run speed (it would take too long because of redundant calculations eg. 0*0).

5.2 Part (b)

In order to be able to conduct the regression, I would take advantage of the *R-package*, *spam*. This package would allow me to take the large matrices of Z and X and identify the non-zero elements in the large matrices. Knowing the location of the non-zero elements within the matrices would then enable a faster computation of the transpose and inverse of Z as well as the product of the first equation to product $\hat{X}$. The same technique can be used to accomplish the computing of $\hat{\beta}$.