STAT 243 PS7

Lei Zhang

November 18, 2017

1 Problem 1

We can calculate the sample standard error based on the m estimates for the coefficient. And then we take the mean of the m standard error from all the simulation. We can then compare the two values and see if they are almost the same or not.

2 Problem 2

Since A is symmetric, we can write $A = \Gamma\Lambda\Gamma^{-1}$, where Γ is an orthogonal matrix with the eigenvectors as the columns and Λ is a diagonal matrix of eigenvalues $\Lambda_{ii} = \lambda_i$. Then, we have

$$\begin{aligned}
\| A \|_2 &= \sup_{z:\|z\|_2=1} \sqrt{(Az)^T(Az)} = \sup_{z:\|z\|_2=1} \sqrt{(z^T A^T Az)} = \sup_{z:\|z\|_2=1} \sqrt{z^T\Gamma\Lambda\Gamma^{-1}\Gamma\Lambda\Gamma^{-1}z} \\
&= \sup_{z:\|z\|_2=1} \sqrt{z^T\Gamma\Lambda^2\Gamma^{-1}z} = \sup_{z:\|z\|_2=1} \sqrt{(\Gamma^T z)^T\Gamma^2(\Gamma^T z)}
\end{aligned}$$

We then set $y = \Gamma^T z$, and note that

$$\| y \|_2 = \| \Gamma^T z \|_2 = y^T y = z^T\Gamma\Gamma^T z = z^T z = \| z \|_2$$

So we have,

$$\| A \|_2 = \sup_{y:\|y\|_2=1} \sqrt{y^T\Gamma^2 y} = \begin{pmatrix} y_1 & .. & y_n \end{pmatrix} \begin{pmatrix} \lambda_1^2 & & \\ & .. & \\ & & \lambda_n^2 \end{pmatrix} \begin{pmatrix} y_1 \\ .. \\ y_n \end{pmatrix}$$

Based on the fact that $\| y \|_2 = 1$, we can conclude that

$$\| A \|_2 = \text{the largest of the absolute values of the eigenvalues of A}$$

3 Problem 3

3.1 Problem a

Let UDV^T be the Singular Value Decomposition of X, where

$$X_{n\times p} = U_{n\times n} D_{k\times k} V^T_{k\times p}$$

$$U_{n\times k} = (\mathbf{u_1}, ...\mathbf{u_k}), V_{p\times k} = (\mathbf{v_1}, ..., \mathbf{v_k}), D_{k\times k} = diag(\lambda_1, ..., \lambda_k)$$

We then have

$$X^T X = (VD^T U^T)(UDV^T) = VD^T DV^T = V \begin{pmatrix} \lambda_1^2 & & \\ & .. & \\ & & \lambda_k^2 \end{pmatrix} V^T$$

$$\begin{aligned} (X^T X)\mathbf{v_i} &= (\mathbf{v_1}, ..., \mathbf{v_k}) \begin{pmatrix} \lambda_1^2 & & \\ & .. & \\ & & \lambda_k^2 \end{pmatrix} \begin{pmatrix} \mathbf{v_1^T} \\ .. \\ \mathbf{v_k^T} \end{pmatrix} \mathbf{v_i} \\ &= (\mathbf{v_1}, ..., \mathbf{v_k}) \begin{pmatrix} \lambda_1^2 & & \\ & .. & \\ & & \lambda_k^2 \end{pmatrix} \begin{pmatrix} 0 \\ .. \\ 1 \\ .. \\ 0 \end{pmatrix} \\ &= (\mathbf{v_1}, ..., \mathbf{v_k}) \begin{pmatrix} 0 \\ .. \\ \lambda_i^2 \\ .. \\ 0 \end{pmatrix} \\ &= \lambda_i^2 \mathbf{v_i} \end{aligned}$$

So we have showed that the right singular vector of X are the eigenvectors of the matrix $X^T X$, and that the eigenvalues of $X^T X$ are the squares of the singular values of X.

We then show that $X^T X$ is positive semi-definite.

$$\forall a \in \mathbb{R}^p, a^T (X^T X) a = (Xa)^T (Xa) = \parallel Xa \parallel_2 \geq 0$$

3.2 Problem b

Suppose the eigendecomposition of Σ is $\Gamma^{-1}\Lambda\Gamma$. We then have $\forall i = 1, ...n$

$$Z\Gamma_{\cdot i} = (\Sigma + cI)\Gamma_{\cdot i} = \Sigma\Gamma_{\cdot i} + c\Gamma_{\cdot i} = \lambda_i\Gamma_{\cdot i} + c\Gamma_{\cdot i} = (\lambda_i + c)\Gamma_{\cdot i}$$

So the eigenvalues of Z are $\lambda_i + c, i = 1, ..., n$, where λ_i is the eigenvalue of Σ

4 Problem 4

4.1 Problem a

As we prove in Problem 3, X^X, is positive semi-definite, and so is $AC^{-1}A^T$. The Pseudo-code is as follow,

1. Perform the Cholesky Decomposition of $C = X^TX$. We can get $C = U^TU$ where U is upper triangular.
2. Calculate $d = X^TY$. Note that it is better to perform multiplications involving vectors first as it may reduce the calculation.
3. Calculate $C^{-1}d$. Note that $C = U^TU$, so we can backsolve the equation to get $C^{-1}d = (U^TU)^{-1}d = U^{-1}(U^{T-1}b)$
4. Calculate $C^{-1}A^T$, $C^{-1}A^T = U^{-1}(U^{T-1}A^T)$
5. Calculate $-AC^{-1}d + b$
6. Calculate $(AC^{-1}A^T)^{-1}(-AC^{-1}d + b)$ using backsolve
7. Calculate $\hat{\beta}$

4.2 Problem b

```
X <- matrix(c(1,0,0,2), nrow = 2, ncol = 2)
Y <- matrix(c(10,11), nrow = 2)
b <- matrix(2, nrow = 1, ncol = 1)
A <- matrix(c(5,2), nrow =1, ncol = 2)
calbetahat <- function(X,Y,b,A){
  C <- crossprod(X)
  #Perform Cholesky Decomposition of C
  U1 <- chol(C)
  d <- crossprod(X,Y)
```

```
  #Backsolve to find C^{-1}d and C^{-1}t(A)
  invC.d <- backsolve(U1, backsolve(U1, d, transpose = TRUE))
  invC.At <- backsolve(U1,backsolve(U1, t(A), transpose=TRUE))
  sumterm <- - A %*% invC.d + b
  inverseterm <- A %*% invC.At
  #Perform the Cholesky Decomposition again
  U2 = chol(inverseterm)
  #calculate betahat
  betahat <- invC.d + invC.At %*% backsolve(U2,backsolve(U2, sumterm, transpose=TRUE))
  return(betahat)
}
calbetahat(X,Y,b,A)

##            [,1]
## [1,] -1.346154
## [2,]  4.365385
```

5 Problem 5

5.1 Problem a

It is because that Z, X and y are pretty large and the calculation is a bit complicated so we can hardly handle this without a huge amount of memory.

5.2 Problem b

$$\begin{aligned}\hat{\beta} &= (\hat{X}^T\hat{X})^{-1}\hat{X}^Ty \\ &= (X^TZ(Z^TZ)^{-1}Z^TX)^{-1}X^TZ(Z^TZ)^{-1}Z^Ty \\ &= (AX^{-1}Ay\end{aligned}$$

where $A = X^TZ(Z^TZ)^{-1}Z^T$

The Pseudo-code is as follow,

1. Perform the Cholesky Decomposition of Z^TZ. We can get $Z^TZ = U^TU$ where U is upper triangular.

2. Calculate $(Z^TZ)^{-1}Z^T$. Note that we can backsolve twice to get the answer

3. Calculate Ay, note that although Z,X,y are pretty large, Ay is a 600×1 vector.

4. Calculate AX, AX is a 600×600 matrix.

5. Calculate the Cholesky decomposition of AX, since $AX = X^T Z(Z^T Z)^{-1} Z^T X$ is positive definite

6. Backsolve twice to find $(AX)^{-1}Ay$, which is $\hat{\beta}$

6 Problem 6

Pseudo-code:

1. Generate Z from rnorm. We then calculate matrix $A = Z^T Z$. A is symmetric, so we have the eigen vectors of A

2. Create a list of maximum positive eigenvalues, each of different magnitudes.

3. In my case, I also set all other eigenvalues in a particular λ matrix to be 1. Note that the condition number which is defined as is the ratio of the absolute values of the largest to smallest eigenvalue under L_2 norm

$$c = \frac{|max_{1 \leq i \leq n} \lambda_i|}{|min_{1 \leq i \leq n} \lambda_i|}$$

is exactly the maximum eigenvalue.

4. Create $\Gamma \Lambda \Gamma^T$ accordingly.

5. Calculate the error in the estimated eigenvalues relative to the known true values.

$$Error = \parallel (trueVector - computedVector) \parallel_2$$

We can see that the error between true eigenvalues and computed eigenvalues increases dramatically after the condition number goes beyond 1e15. Furthermore, we see that when the condiiton number is larger than 1e16, the matrix may not be numerically positive definite. We can see this from the fact that computed eigenvalues are not all strictly positive.

```
n <- 100#matrix size
Z <- matrix(rnorm(n^2), nrow = n, ncol = n)
A <- crossprod(Z)
eigen <- eigen(A)
```

```r
eigenVectors <- eigen$vectors
# These are my condition numbers since my smallest eigenvalue is always 1
maxEvalues <- c(0.1,1,10,100,1000,10000,1e5,1e6,1e7,
1e8, 1e9, 1e10, 1e11, 1e12, 1e13, 1e14,1e15,1e16,1e17)
error <- vector("list", length(maxEvalues))

for (i in 1:length(maxEvalues)) {
#Create the true lambda matrix, all the diagonal elements are one
#except the [1,1] element, which loop through the maxEvalue vector
  Lambda <- diag(n)
  Lambda[1,1] <- maxEvalues[i]
  trueEvalues <- sort(diag(Lambda))
#Calculate the new matrix \Gamma^T\Lambda\Gamma
  M <- eigenVectors %*% Lambda %*% t(eigenVectors)
#Computed eigenvalues
  computedEvalues <- sort(eigen(M)$values)
#print out the maximum eigenvalue when the matrix is not numerically
  #positive definite
  if (length(computedEvalues[computedEvalues <= 0]) != 0){
    print(maxEvalues[i])
  }
  #Calculate the L2 norm of the error
  error[i] = sum((trueEvalues - computedEvalues)^2)
  i = i+1
}
##[1] 1e+16
##[1] 1e+17
library(ggplot2)
data <- data.frame(conditionNum = maxEvalues, error=unlist(error))
p1 <- ggplot(df, aes(conditionNum)) +
geom_line(aes(y=error, colour = "Error")) +
labs(x="Condition Number") +
labs(y="Error") +
labs(title="Condition Number and eigenValue Errors")
print(p1)
```