DATA 605 - Assignment 2

Joshua Sturm

02/05/2018

Problem Set 1

1. Show that $A^TA \neq AA^T$ in general. (Proof and demonstration.)

Solution

Let $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$. Then $A^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$.

The product of AA^T will be: $\begin{bmatrix} a \times a + b \times b & a \times c + b \times d \\ c \times a + d \times b & c \times c + d \times d \end{bmatrix}$.

The product of A^TA will be: $\begin{bmatrix} a \times a + c \times c & a \times b + c \times d \\ a \times b + c \times d & b \times b + d \times d \end{bmatrix}$.

Thus, we can see that matrix multiplication does not have the commutative property.

In the event that A is not a square matrix, then the proof is trivial. Say B is an $m \times n$ matrix, then B^T will be an $n \times m$ matrix. BB^T will thus be an $m \times m$ matrix, which is a square matrix.

2. For a special type of square matrix A, we get $A^TA = AA^T$. Under what conditions could this be true? (Hint: The identity matrix I is an example of such a matrix.)

Solution

Theorem: For any matrix A, AA^T and A^TA are symmetric matrices.
The product of two symmetric matrices is symmetric only if both matrices commute, i.e. $AA^T = A^TA$. (https://en.wikipedia.org/wiki/Symmetric_matrix#Properties)/ This is only possible when the matrix is invertible.

Problem Set 3

Write an R function to factorize a square matrix A into LU or LDU, whichever you prefer. You don't have to worry about permuting rows of A and you can assume that A is less than 5x5, if you need to hard-code any variables in your code

```
# Assume the user inputs a square matrix.

factorize <- function(mat){
  L <- diag(nrow(mat))    # Create lower diagonal matrix of the same size
  U <- mat

  # If the leading entry [1,1] is not > 1, swap it with a row that has a pivot
  if (U[1,1] == 0){
    for (i in U[2,1]:U[nrow,1]){
```

```r
      if (U[i,1] != 0){
        U[1,] <- U[i,]
      }
    }
  }
  # Loop through n-1 columns, n rows, get the multipler,
  # Multiply to get the Upper Triangular matrix, and plug
  # the multiplier into the Lower Triangular Matrix
  for (k in 1:(nrow(U)-1)){
    for (j in (k+1):(nrow(U))){
      if (U[j,k] != 0){
        mult <- U[j,k] / U[k,k]
        U[j,] <- U[j,] - (mult * U[k,])
        L[j,k] <- mult
      }
    }
  }

  factored.matrix <- list(L, U)

  return(factored.matrix)
}
```