Data 605 - HW2

Avery Davidowitz

2022-11-25

Problem set 1

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

We can easily prove this by contradiction. Let us assume that $A^T A = AA^T$ for any matrix A and we will find that this introduces a contradiction. Let $A = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$ which is 3x1, then $A^T = \begin{bmatrix} 2 & 3 & 1 \end{bmatrix}$ which is 1x3. By the definition of matrix multiplication $A^T A$ will result in a 1x1 matrix and AA^T will result in a 3x3 matrix. Therefore, they cannot be equal and we now have a contradiction. This proves that $A^T A \neq AA^T \quad \forall A$

Demonstrating in R:

```
A = matrix(c(2,3,1),
             3, byrow=TRUE)
A
```

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

```
t(A)
```

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

```
t(A) %*% A
```

```
##      [,1]
## [1,]   14
```

```
A %*% t(A)
```

```
##      [,1] [,2] [,3]
## [1,]    4    6    2
## [2,]    6    9    3
## [3,]    2    3    1
```

2 For a special type of square matrix A, we get AT A = AAT . Under what conditions could this be true? (Hint: The Identity matrix I is an example of such a matrix).

This would be true in general for a symmetric matrix where by definition $A = A^T$

Problem set 2

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.

```
lu_decomp <- function (A) {
  ### returns the LU decomposition for a matrix A with an assumption
  ### that no permutation matrix P is required

  # check for square dimension
  n <- dim(A)
  if (n[1] != n[2]) stop("must be a square")
  n <- n[1]
  L <- diag(n)
  U <- A
  for(i in 2:n){
    for(j in 1:(i-1)){
      L[i,j] <- U[i,j] / U[j,j] #constants used to zero out the first column of A
      U[i,] <- U[i,] - L[i,j] * U[j,] #first row of U = zero out the first entry of rows below it
    }
  }
  return(list(L,U))
  }
```

```
B <- matrix(c(1, 2, -7, -1, -1, 1, 2, 1, 5), 3)
lu <- lu_decomp(B)
lu
```

```
## [[1]]
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    2    1    0
## [3,]   -7   -6    1
##
## [[2]]
##      [,1] [,2] [,3]
## [1,]    1   -1    2
## [2,]    0    1   -3
## [3,]    0    0    1
```