DATA605 Assignment2

Harpreet Shoker

February 10, 2018

(1) **Show that in general:**

$$A^T A \neq AA^T$$

Proof and demonstration

$$A = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

$$SoA^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

Now we need to calculate

$$A.A^T$$

$$\begin{bmatrix} a^2+b^2 & ac+bd \\ ac+bd & c^2+d^2 \end{bmatrix}$$

Now calculating

$$A^T.A$$

$$\begin{bmatrix} a^2+c^2 & bc+cd \\ ab+cd & b^2+d^2s \end{bmatrix}$$

From the above two results we can say

$$A.A^T \neq A^T.A$$

Here is an example to prove the above

Abstract

```
  A <- matrix(c(1,2,3,4),nrow = 2,ncol = 2)
A

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

```
A_T <- t(A)
A_T
```

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

```
X <- A %*% A_T
X
```

```
##      [,1] [,2]
## [1,]   10   14
## [2,]   14   20
```

```
Y <- A_T %*% A
Y
```

```
##      [,1] [,2]
## [1,]    5   11
## [2,]   11   25
```

From the above example again we can say

$$A.A^T \neq A^T.A$$

(2) **For a special type of square matrix A,**

$$A^T A \neq AA^T$$

Under what conditions could this be true?

For the above equation to be satisfied one is matrix could be identity matrix and other could be matrix should be symmetric.We will prove here using symmetric matrix

Abstract

```
   A <- matrix(c(1,7,3,7,4,-5,3,-5,6),nrow = 3,ncol = 3)
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    7    3
## [2,]    7    4   -5
## [3,]    3   -5    6
```

```
A_T <- t(A)
A_T
```

```
##      [,1] [,2] [,3]
## [1,]    1    7    3
## [2,]    7    4   -5
## [3,]    3   -5    6
```

```
X <- A %*% A_T
X
```

```
##      [,1] [,2] [,3]
## [1,]   59   20  -14
## [2,]   20   90  -29
## [3,]  -14  -29   70
```

```
Y <- A_T %*% A
Y
```

```
##      [,1] [,2] [,3]
## [1,]   59   20  -14
## [2,]   20   90  -29
## [3,]  -14  -29   70
```

From the above example of symmetric matrix again we can say

$$A.A^T isequalto A^T.A$$

(3) **Matrix factorization is a very important problem. There are supercomputers built just to do matrix factorizations. Every second you are on an airplane, matrices are being factorized. Radars that track flights use a technique called Kalman filtering. At the heart of Kalman Filtering is a Matrix Factorization operation. Kalman Filters are solving linear systems of equations when they track your flight using radars. Write an R function to factorize a square matrix A into LU or LDU, whichever you prefer**

Abstract

```
 Matrix_Factorization <- function(A){
  if(nrow(A) != ncol(A)){ #checking whether matrix is square or not
    stop("Matrix is not a square matrix")
  }
  L <- diag(nrow(A)) # Generates the lower triangular matrix
  for (i in 2:nrow(A)){ # i starts from row 2
    for (j in 1:(i-1)){ # j Columnns will not go through the last col-
umn
      if (A[i,j] == 0){
        stop("Factorization cannot be done as 0 in pivot")
      }
      L[i,j] <- (A[i,j]/A[j,j]) # calculating L matrix values
      A[i,] <- A[i,] - (L[i,j] * A[j,]) #  Then treating A as U

    }
  }
  print("Lower matrix:")
  print(L)
  print("Upper matrix:")
  print(A)
}
```

```r
m3 <- matrix(c(2,5,9,2,9,1,3,10,2), ncol = 3, nrow=3)
Matrix_Factorization(m3)
```

```
## [1] "Lower matrix:"
##      [,1] [,2] [,3]
## [1,]  1.0    0    0
## [2,]  2.5    1    0
## [3,]  4.5   -2    1
## [1] "Upper matrix:"
##      [,1] [,2] [,3]
## [1,]    2    2  3.0
## [2,]    0    4  2.5
## [3,]    0    0 -6.5
```

```r
m2 <- matrix(c(1,2,3,4,5,6),nrow = 3,ncol = 2)
Matrix_Factorization(m2)
```

```
## Error in Matrix_Factorization(m2):  Matrix is not a square matrix
```