Week_2 605 assignment

Salma Elshahawy

1/14/2020

Problem set_1

(1) Show that "$A^T A \neq AA^T$" in general.

```
# creating a matrix A
A <- matrix(c(1,2,3,4,5,6,7,8,9), nrow = 3)
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
# get A transpose and assign it to At
At <- t(A)
At
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

```
# mutliply matrix A by it's transpose A*t(A)
left_side <- A %*% At
left_side
```

```
##      [,1] [,2] [,3]
## [1,]   66   78   90
## [2,]   78   93  108
## [3,]   90  108  126
```

```
# mutliply matrix transpose A by the original matrix t(A)*A
right_side <- At %*% A
right_side
```

```
##      [,1] [,2] [,3]
## [1,]   14   32   50
## [2,]   32   77  122
## [3,]   50  122  194
```

```
# check if the two sides are in equilibrium state
left_side == right_side
```

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

Both sides are not equal, so in general $A^TA \quad \neq \quad AA^T$

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

Please typeset your response using LaTeX/SWeave mode in RStudio. If you do it in paper, please either scan or take a picture of the work and submit it. Please en- sure that your image is legible and that your submissions are named using your first initial, last name, assignment and problem set within the assignment.

```r
# construct a diagonal matrix
#
A <- matrix(c(1,0,0,0,5,0,0,0,9), nrow = 3)
A <- 5 * A # multiply by a scaler
A
```

```
##      [,1] [,2] [,3]
## [1,]    5    0    0
## [2,]    0   25    0
## [3,]    0    0   45
```

```r
A%*%t(A) == t(A)%*%A
```

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

I Observed that when a matrix is symmetric, the matrix is equal to its transpose, $A^TA \quad = \quad AA^T$

The conditions:

1- Be a square matrix $A_{i\times j} \quad = \quad A^T_{j\times i}$

2- The upper and lower triangle are equal to zero $\begin{matrix} a_{ij} & 0 & 0 \\ 0 & a_{ij} & 0 \\ 0 & 0 & a_{ij} \end{matrix}$

Problem set_2

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.

Please submit your response in an R Markdown document using our class naming convention. 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. If you doing the entire assignment in R, then please submit only one markdown document for both the problems.

```r
a <- matrix(c(2,4,-2,1,-1,5,3,3,5), ncol = 3)
a
```

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

```r
get_UL <- function(a) {
  U = a
  L = diag(nrow(a))
  n = nrow(a)
  for (i in 1:n) {
    k = seq(2, n)
    for (j in k) {
      if(j > i) {
        # get the multiplier and add it to the L matrix
        s = U[[j,i]]/U[[i,i]]
        L[j,i] = s
        # reduce by reduction and shuffle to the U matrix
        U[j,] = U[j,] - s * U[i,]
      }
    }
  }
  return(list(U = U, L = L))
}

value <- get_UL(a)
U_matrix <- value$U
U_matrix
```

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

```r
L_matrix <- value$L
L_matrix
```

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