HW2

FUNDAMENTALS OF COMPUTATIONAL MATHEMATICS

Zachary Palmore

2/4/2021

Directions

1. Problem set 1

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

(2) For a special type of square matrix A, we get $A^TA \neq AA^T$. 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 mode in RStudio. If you do it in paper, please either scan or take a picture of the work and submit it. Please ensure that your image is legible and that your submissions are named using your first initial, last name, assignment and problem set within the assignment. E.g. LFulton_Assignment2_PS1.png

2. 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, E.g. LFulton_Assignment2_PS2.png

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.

Problem Set 1

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

As a demonstration, the matrix A and its transposed form A^T were created below. The products of A^TA and AA^T were also computed using this chunk of code. To keep it simple, the matrix A repeated the number 2 for all entries in matrix A with three columns and two rows. This is also used to better show how matrix multiplication differs when the dimensions of the matrices also differ.

```r
A <- matrix(c(2), ncol = 3, nrow = 2, byrow = TRUE)
At <- t(A)
prod_AAt <- A %*% At
prod_AtA <- At %*% A
B <- matrix(c(2), ncol = 2, nrow = 2)
Bt <- t(B)
BBt <- B %*% Bt
BtB <- Bt %*% B
```

The matrices of A and its transposed form A^T are shown here with the operation of multiplication.

$$\begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \times \begin{bmatrix} 2 & 2 \\ 2 & 2 \\ 2 & 2 \end{bmatrix}$$

Results from that operation are shown here:

$$\begin{bmatrix} 12 & 12 \\ 12 & 12 \end{bmatrix}$$

Notice that in the matrix A, there are two rows and three columns. All values are the number 2 as entered. In the matrix A^T the dimensions of those rows and columns change, but all the entries remain two. Instead of two rows and three columns this transposed matrix displays three columns with two rows.

The results of this operation produce a matrix that is two rows by two columns and the values are different. All values are the number 12. We repeat this operation after rearranging the order of the matrix and its transposed form. The operation of the matrix's transposed form A^T and its not transposed form A is set up the same.

$$\begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \times \begin{bmatrix} 2 & 2 \\ 2 & 2 \\ 2 & 2 \end{bmatrix}$$

Results from that operation are shown here:

$$\begin{bmatrix} 8 & 8 & 8 \\ 8 & 8 & 8 \\ 8 & 8 & 8 \end{bmatrix}$$

With this rearranging of order, the results change. Notice that we now have a matrix with three rows and three columns all with the value 8. This is no coincidence.

If in general $A^TA = AA^T$ then we would expect two identical matrices. But based on these results, we can see this did not occur. The dimensions of both resultant matrices are different and so are its products. We can look at this a bit closer.

For example, we see that when matrix A, with dimensions A_{n*k}, is transposed it displays a matrix of At_{k*n}, where $_{n*k}$ contains the number of rows and columns respectively. This occurred when $n \neq k$. From this we can also gather that where $n \neq k$ then the following are true of the matrix A and its dimensions A_{n*k}:

$$Where\ n \neq k$$
$$A^T = k * n$$
$$AA^T = k * k$$
$$A^TA = n * n$$

These properties appear to remain true for any matrix where $n \neq k$. However, when we compare results of the same operations on a 2x2 matrix, B, which also has all values of 2 entered into it, these properties do not hold. Where $n = k$ the matrix B and B^T produce a square matrix with the dimensions of $n * K$ and the results are identical. This is demonstrated with the product of matrix BB^T and B^TB respectively below.

$$\begin{bmatrix} 8 & 8 \\ 8 & 8 \end{bmatrix} \begin{bmatrix} 8 & 8 \\ 8 & 8 \end{bmatrix}$$

(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). Please typeset your response using LaTeX mode in RStudio. If you do it in paper, please either scan or take a picture of the work and submit it. Please ensure that your image is legible and that your submissions are named using your first initial, last name, assignment and problem set within the assignment. E.g. LFulton_Assignment2_PS1.png

If the square matrix had values that when ordered were identical along the diagonals of the matrix then transposing the matrix would display the same result. This can be proven with examples.

In this chunk we create a 3 by 3 matrix M with the values 1, 2, 3... until 9. We then transpose it to create matrix M^T. We then repeat this process of creating a matrix I and transposing it to create I^T with the even integer values 2, 4, 6, 4, 2, 6, 6, 6, and 8. Then we visualize.

```
M <- matrix(c(1:9), ncol = 3, nrow = 3, byrow = TRUE)
Mt <- t(M)
I <-  matrix(c(2,4,6,4,2,6,6,6,8), ncol = 3, nrow = 3, byrow = TRUE)
It <- t(I)
```

For the first matrix with values 1-9 we produce the following matrix M and its transposed form Mt.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} OR \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

Notice how the columns and rows are flipped. Transposing causes the rows become the columns and columns become rows. When evaluating if this satisfies the statement $A^TA \neq AA^T$ we can see it is false. The transposed matrix M^T is not identical to that of M.

For the second matrix with the even integer values 2, 4, 6, 4, 2, 6, 6, 6, and 8 we produce the following matrix I and its transposed form M^T.

$$\begin{bmatrix} 2 & 4 & 6 \\ 4 & 2 & 6 \\ 6 & 6 & 8 \end{bmatrix} OR \begin{bmatrix} 2 & 4 & 6 \\ 4 & 2 & 6 \\ 6 & 6 & 8 \end{bmatrix}$$

Here, the results are identical. The matrix I^T is said to be equal to matrix I. In other words this special type of square matrix satisfies the conditions for the statement $A^TA = AA^T$ to be true. But how? Because this square matrix has symmetry along its diagonal. That is, for each value in the matrix there is identical value if the matrix were folded in half diagonally from the corners in either direction.

Problem Set 2

(1) 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, E.g. LFulton_Assignment2_PS2.png

The function, factorized into lower and upper (LU) matrices is written below. In it, to decompose the matrix A its dimensons were found and evaluated to determine if the matrix was indeed square (equal on both sides, length and width), and if it was not to return NA as a LU decomposition cannot be performed. Next, pending those prior conditions were met, we stored the variables associated with matrix A including the length of one side of the square matrix as n. An empty matrix L was filled with values based on the length n. What followed was the use of a fraction to cancel out the value in the matrix and produce a zero for each matrix value in that is lower than the diagonal of the matrix. Using this method the value in row two of column one was zeroed out and the factor used to find this stored in L. Remaining factors after the operation were stored in U and the resultant lists created two matrices from matrix A.

```
LU <- function(A) {
      if (dim(A)[1]!=dim(A)[2]) {
          return(NA)}
  U <- A
  n <- dim(A)[1]
  L <- diag(n)
  if (n==1) {
    return(list(L,U))
}
  for(i in 2:n) {
    for(j in 1:(i-1)) {
      factor <-
        -U[i,j] / U[j,j]
      U[i, ] <-
        round(factor * U[j, ] + U[i, ], digits = 1)
      L[i,j] <-
        round(-(factor), digits = 1)} }
  return(list(L,U))
}
```

With the function created, it is tested on matrix Z as well as its transposed form, Z^T, to evaluate. Values for the test were randomly selected numbers between 0 and 10. The creation of the test matrix, its transposed form, and the results of their LU decomposition are labeled as L and U for the not transposed, L^T and U^T for the transposed.

```
Z <- matrix(c(-1,5,8,10,-7,2,-4,0,1), ncol = 3, nrow = 3, byrow = TRUE)
Zt <- t(Z)
LUZ <- LU(Z)
LUZt <- LU(Zt)
```

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -10 & 1 & 0 \\ 4 & -0.5 & 1 \end{bmatrix} U = \begin{bmatrix} -1 & 5 & 8 \\ 0 & 43 & 82 \\ 0 & 0 & 7.1 \end{bmatrix}$$

$$L^T = \begin{bmatrix} 1 & 0 & 0 \\ -5 & 1 & 0 \\ -8 & 1.9 & 1 \end{bmatrix} U^T = \begin{bmatrix} -1 & 10 & -4 \\ 0 & 43 & -20 \\ 0 & 0 & 7.1 \end{bmatrix}$$

If this were to be tested by hand the matrix Z would be broken down in steps. At each step, the factor used to multiply the top row to cancel out the denominator and create a zero in the upper portion of LU, is saved. This process is known as putting the matrix in reduced row echelon form. Once completed, the inverse of those factors are entered as values in the lower portion of the LU creating the upper matrix. These steps can be demonstrated.

We start with Z.

$$\begin{bmatrix} -1 & 5 & 8 \\ 10 & -7 & 2 \\ -4 & 0 & 1 \end{bmatrix}$$

It is a 3x3 matrix so the process should only take three factors to change the bottom left section of the matrix U to zeros and place their inverses into the upper matrix L. We being by multiplying the first row by a factor of 10 and adding it to the second row to produce a zero. This creates matrix Z_1.

$$\begin{bmatrix} -1 & 5 & 8 \\ 0 & 43 & 82 \\ -4 & 0 & 1 \end{bmatrix}$$

Now we have the first factor 10 and we continue to put the matrix in reduced row echelon form. The next step is to multiply the values in the first row by a factor of -4 to add to the third row to get zero. This forms Z_2.

$$\begin{bmatrix} -1 & 5 & 8 \\ 0 & 43 & 82 \\ 0 & -20 & -31 \end{bmatrix}$$

Our next factor is -4 and the process is repeated. To get a zero in the second column of the third row we multiply the first row by 20/43 and add it to the third row. Our result is Z_3.

$$\begin{bmatrix} -1 & 5 & 8 \\ 0 & 43 & 82 \\ 0 & 0 & 7.1 \end{bmatrix}$$

The last factor was 20/43 or approximately 7.1. This completes the factorization of matrix Z. This same process was used to create the function LU which produced those matricies above. The transposed versions should share the same diagonal values if it functions properly and for matrix Z, it does.