IS605 - Assignment 2

Suman K Polavarapu

Saturday, September 05, 2015

Problem Set 1 :

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

Let A be an m x n matrix , so A^T is an n x m matrix. By multiplying A by A^T produces an m x m matrix. And multiplying A^T with A would produce n x n matrix. So, the matrices would clearly differ in dimension, when m != n.

Lets check a case where m == n.

$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$

$A^T = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$

$AA^T = \begin{bmatrix} a^2+b^2+c^2 & ad+be+cf & ag+bh+ci \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}$

However, $A^TA = \begin{bmatrix} a^2+d^2+g^2 & ab+de+gh & ac+df+gi \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}$

Notice the mismatch of the dot products of the first row elements itself. So, generally $A^TA \neq AA^T$

(2) For a special type of square matrix A, we get $A^TA = AA^T$. Under what conditions could this be true?

$A^TA = AA^T$ is true , if a matrix is a square matrix and all of its elements are same, Or, if the matrix is an identity matrix (/or a scalar multiples of it).

```
A <- matrix(c(5,5,5,5,5,5,5,5,5), nrow=3, byrow=T)
A
```

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

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

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

Similarly, a diagonal matrix.

```
A <- matrix(c(5,0,0,0,5,0,0,0,5), nrow=3, byrow=T)
A
```

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

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

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

Problem Set 2 :

Write an R function to factorize a sqaure matrix A into LU or LDU.

```
###################
#Decompose/Factorize a given matrix into L and U.
#Inputs: A - matrix, which need to be factorized.
#        We will apply the elimination steps to this, which eventually be our U.
#Output: A list of L and U.
###################

FactorizeLU <- function(U) {

  rows = nrow(U)
  cols = ncol(U)

  if ( rows != cols) {
    print("*** Given Matrix is NOT a square matrix !!! *** ")
  }

  #Identity matrix, to store the multipliers used in elimination, this would be
  # eventually our lower triangular matrix.
  L <- diag(rows)

  #Start eliminating the E21, (/E31, E32 etc..) from the matrix U,
  #and in that process capture the E (elimination steps) as well
  for(i in 2:rows) {

    for(j in 1:(i-1)) {

      #Generate Elimination matrix.
      #For row2 , it would be E21, For row3, it would be E31, E32 etc..
      E <- diag(rows)
      E[i, j] <- -(U[i,j]/U[j,j])  #-(multiplier), for E21, it would be - U[2,1]/U[1,1]
```

```
      #Eliminate the elements from Upper triangular matrix.Because,  U = (E32E31E21) A
      U <- E %*% U

      #keep the elimination step in L. (the multiplier used in the elimination)
      #because, L <- solve(E21) %*% solve(E31) %*% solve(E32)
      L <- L %*% solve(E)

    } # for each column
  } # for each row

  return(list('L'=L,'U'=U))  #return L as L , and U as U attribute.

}
```

```
(A <- matrix(c(2,1,6,5), nrow=2))
```

Client calls:

```
##      [,1] [,2]
## [1,]    2    6
## [2,]    1    5
```

```
(res <- FactorizeLU(A))
```

```
## $L
##      [,1] [,2]
## [1,]  1.0    0
## [2,]  0.5    1
##
## $U
##      [,1] [,2]
## [1,]    2    6
## [2,]    0    2
```

```
(res$L %*% res$U == A)
```

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

```
(A <- matrix(c(1, 2, 1, 3, 4, 1, 5, 7, 0), nrow=3))
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    7
## [3,]    1    1    0
```

```r
(res <- FactorizeLU(A))
```

```
## $L
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    2    1    0
## [3,]    1    1    1
##
## $U
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    0   -2   -3
## [3,]    0    0   -2
```

```r
(res$L %*% res$U == A)
```

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

LU Decomposition using package - matrixcalc

```r
library(matrixcalc)
A <- matrix( c (1, 2, 1, 3, 4, 1, 5, 7, 0 ), nrow=3)
luA <- lu.decomposition( A )
(L <- luA$L)
```

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

```r
(U <- luA$U)
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    0   -2   -3
## [3,]    0    0   -2
```

```r
(luA$L %*% luA$U == A)
```

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