Trace, Determinant, Factorization of MatricesIS Assignment 2

Sharon Morris

2/7/2017

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

The commutative law shows that $AB \neq BA$

```
#Create matrix
A <- matrix(c(0,0,0,1,2,3,2,4,6), nrow = 3, byrow = T)
A
```

```
##      [,1] [,2] [,3]
## [1,]    0    0    0
## [2,]    1    2    3
## [3,]    2    4    6
```

```
t(A)  # transpose the martrix
```

```
##      [,1] [,2] [,3]
## [1,]    0    1    2
## [2,]    0    2    4
## [3,]    0    3    6
```

```
AtA <- t(A) %*% A  # t(A) times non-transposed matrix
AtA
```

```
##      [,1] [,2] [,3]
## [1,]    5   10   15
## [2,]   10   20   30
## [3,]   15   30   45
```

```
AAt <- A %*% t(A)  #non-transposed matrix times t(A)
AAt
```

```
##      [,1] [,2] [,3]
## [1,]    0    0    0
## [2,]    0   14   28
## [3,]    0   28   56
```

```
AtA == AAt  # Are the matrices equal
```

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

(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).

```
A2 <- matrix(c(2,1,3,1,0,1,3,1,2), nrow = 3, byrow = T)
A2
```

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

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

```
t(A2) # transpose matrix
```

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

```
AA2 <- t(A2) %*% A2
AA2
```

```
##      [,1] [,2] [,3]
## [1,]   14    5   13
## [2,]    5    2    5
## [3,]   13    5   14
```

```
A2A <- A2 %*% t(A2)
A2A
```

```
##      [,1] [,2] [,3]
## [1,]   14    5   13
## [2,]    5    2    5
## [3,]   13    5   14
```

```
AA2 == A2A
```

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

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.

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.

```
functA  <- function(A) {
  dimA =dim(A)[1]
  dimA

  idMatrix <- diag( dim(A)[1]) # crete identity matrix
  idMatrix

  iParameters = idMatrix   # initialize parameters
  iParameters[2,1] <- -A[2,1]/A[1,1]
  lParameters = iParameters

  iParameters = iParameters %*% A # inialize upper matrix
  U=lParameters
  i=1

  U= solve(U) # initialize lower matrix
```

```
   # loop through the column martix
  for  (j in 1:(dimA-i)) {
    #y <- diag( dim(A)[1])
    for  (i in 3:dimA) {
      # check if pivot element is 0.
      if (i != j) {
        if (iParameters[i,j] != 0 ) {
          if (iParameters[j,j] < 0)  idMatrix[i,j] <- -iParameters[i,j] /     iParameters[j,j]
          if (iParameters[j,j] > 0  )  idMatrix[i,j] <- -iParameters[i,j]/ iParameters[j,j]

          iParameters= idMatrix%*%iParameters # holder upper matrix
          U= U %*%solve(idMatrix) # holder lower matrix

          print(iParameters)
          print(U)
          idMatrix <- diag( dim(A)[1]) # reset to identity matrix

        }

      }

    }
  }
  print(iParameters)
  print(U)
  U=iParameters
  s <- U %*% U
  print (s)
  print(A)
  #return(all.equal(s,A))
}
```

```
# Test 3x3 matrix

A <- matrix(c(2,1,0,1,2,1,0,1,2),nrow=3, byrow=TRUE)
functA(A)
```

```
##      [,1] [,2]     [,3]
## [1,]    2  1.0 0.000000
## [2,]    0  1.5 1.000000
## [3,]    0  0.0 1.333333
##      [,1]      [,2] [,3]
## [1,]  1.0 0.0000000    0
## [2,]  0.5 1.0000000    0
## [3,]  0.0 0.6666667    1
##      [,1] [,2]     [,3]
## [1,]    2  1.0 0.000000
## [2,]    0  1.5 1.000000
## [3,]    0  0.0 1.333333
##      [,1]      [,2] [,3]
## [1,]  1.0 0.0000000    0
## [2,]  0.5 1.0000000    0
## [3,]  0.0 0.6666667    1
##      [,1] [,2]     [,3]
```

```
## [1,]    4 3.50 1.000000
## [2,]    0 2.25 2.833333
## [3,]    0 0.00 1.777778
##      [,1] [,2] [,3]
## [1,]    2    1    0
## [2,]    1    2    1
## [3,]    0    1    2
```

```
# Test 4x4 matrix
A <- matrix(c(1,1,1,1,1,2,3,4,1,3,6,10,1,4,10,20),nrow=4, byrow = T)
functA(A)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    0    1    2    3
## [3,]    0    2    5    9
## [4,]    1    4   10   20
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    1    1    0    0
## [3,]    1    0    1    0
## [4,]    0    0    0    1
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    0    1    2    3
## [3,]    0    2    5    9
## [4,]    0    3    9   19
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    1    1    0    0
## [3,]    1    0    1    0
## [4,]    1    0    0    1
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    0    1    2    3
## [3,]    0    0    1    3
## [4,]    0    3    9   19
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    1    1    0    0
## [3,]    1    2    1    0
## [4,]    1    0    0    1
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    0    1    2    3
## [3,]    0    0    1    3
## [4,]    0    0    3   10
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    1    1    0    0
## [3,]    1    2    1    0
## [4,]    1    3    0    1
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    0    1    2    3
```

```
## [3,]    0    0    1    3
## [4,]    0    0    0    1
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    1    1    0    0
## [3,]    1    2    1    0
## [4,]    1    3    3    1
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    0    1    2    3
## [3,]    0    0    1    3
## [4,]    0    0    0    1
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    1    1    0    0
## [3,]    1    2    1    0
## [4,]    1    3    3    1
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    4    8
## [2,]    0    1    4   12
## [3,]    0    0    1    6
## [4,]    0    0    0    1
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    1    2    3    4
## [3,]    1    3    6   10
## [4,]    1    4   10   20
```