HMW 2-Problem Set 2

Omer Ozeren

Table of Contents

Write an R function to factorize a square matrix *A* into LU or LDU, whichever you prefer.

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

Function Set-Up

```
lu_function <- function(m) {
    mDims <- dim(m)

    # check for square matrix
    if (mDims[1] != mDims[2])
        return(NA)

    U <- m
    n <- mDims[1]
    L <- diag(n)

    # if dim is 1, the U=A and L=[1]
    if (n == 1)
        return(list(L, U))

    # loop through lower triangle determine multiplier
    for (i in 2:n) {
        for (j in 1:(i - 1)) {
            multiplier <- -U[i, j]/U[j, j]
            U[i, ] <- multiplier * U[j, ] + U[i, ]
            L[i, j] <- -multiplier
        }
    }
```

```
    return(list(L, U))
}
```

Function Example

```
m1 <- matrix(seq(1, 9), nrow = 3)
m2 <- matrix(sample(1:100, 9, replace = T), nrow = 3)
ms <- list(m1, m2)
lus <- ms %>% map(lu_function) %>% print
```

```
## [[1]]
## [[1]][[1]]
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    2    1    0
## [3,]    3    2    1
##
## [[1]][[2]]
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    0   -3   -6
## [3,]    0    0    0
##
##
## [[2]]
## [[2]][[1]]
##           [,1]      [,2] [,3]
## [1,] 1.0000000  0.000000    0
## [2,] 1.3333333  1.000000    0
## [3,] 0.6515152 -2.064935    1
##
## [[2]][[2]]
##      [,1]          [,2]      [,3]
## [1,]   66  6.000000e+01   84.0000
## [2,]    0 -1.400000e+01  -68.0000
## [3,]    0  3.552714e-15 -139.1429
```

```
m1 == lus[[1]][[1]] %*% lus[[1]][[2]]
```

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

```
m2 == lus[[2]][[1]] %*% lus[[2]][[2]]
```

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