Factorizacion LU

Uriel Paluch

14/9/2021

La factorización es especialmente útil cuando se tiene la forma $A = LU$, donde L es triangular inferior y U es triangular superior. Resolver $Ax = b$ es equivalente a resolver $LUx = b$

Suponga que A se ha factorizado en la forma triangular $A = LU$. Entonces podemos resulver x con mayor facilidad a través de un proceso de dos pasos:
1. Primero, hacemos $y = Ux$ y resolvemos el sistema triangular superior $Ly = b$ para y.
2. Una vez que conocemos y, resolvemos el sistema triangular superior $Ux = y$.

Teorema:

Si la eliminación gaussiana se puede realizar en el sistema lineal $Ax = b$ sin intercamios de fila, entonces la matriz A se puede factorizar en el producto de una matriz triangular inferior L y una matriz triangular superior U, es decir, $A = LU$, donde $m_{ji} = a_{ji}/a_{ii}$

$$U = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n}^{(n-1)} \\ 0 & \cdots & 0 & a_{nn}^{(n)} \end{bmatrix}, \quad \text{y} \quad L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ m_{n1} & \cdots & m_{n,n-1} & 1 \end{bmatrix}$$

Clarificando un poco lo que menciona el libro: $A = LU$, por ende, $Ax = LUx$. Teniendo en cuenta el paso 1: $Ly = b$. Resolvemos y. Ahora, $Ux = y$, y despejamos x.

```r
factorizacionLU <- function(matriz_coeficientes, vector_resultados){

  n_incognitas <- nrow(vector_resultados)

  #Calculo la matriz U
  U <- matriz_coeficientes
  for (i in 1:(n_incognitas-1)) {
    #La hago triangular inferior
    for (k in (i+1):n_incognitas) {
      U[k,] <- U[k,] - (U[k,i]/U[i,i]) * U[i,]
    }
  }

  #Calculo la matriz L
  L <- matriz_coeficientes
  #Hago unos en la diagonal principal
  for (i in (n_incognitas-1):1) {
```

```r
    L[i,] <- L[i,]/L[i,i]

    #La hago triangular inferior
    for (k in n_incognitas:(i-1)) {
      L[k,] <- L[k,] + L[k,i] * L[i,]
    }
  }

  #print(matriz_coeficientes)
  #print(U)
  print(L)

}

A <- matrix(c(1, 1, 0, 3,
              2, 1, -1, 1,
              3, -1, -1, 2,
              -1, 2, 3, -1), nrow = 4, ncol = 4, byrow = TRUE)

b <- matrix(c(8, 7, 14, -7), nrow = 4, ncol = 1, byrow = TRUE)

factorizacionLU(matriz_coeficientes = A, vector_resultados = b)
```

```
##      [,1] [,2] [,3]        [,4]
## [1,]    2 -0.4 -0.8   0.9333333
## [2,]  -32  5.2 12.4 -17.4666667
## [3,]  -44  8.4 16.8 -24.2666667
## [4,] -100 20.0 41.0 -55.3333333
```

```r
LU <- function(matriz_coeficientes){
  n_incognitas = nrow(matriz_coeficientes)

  L <- matrix(rep(0, times = n_incognitas^2), nrow = n_incognitas, ncol = n_incognitas, byrow = TRUE)
  U <- matrix(rep(0, times = n_incognitas^2), nrow = n_incognitas, ncol = n_incognitas, byrow = TRUE)

  # -------- PASO 1
  for (i in 1:(n_incognitas-1)) {
    #Completo con 1 la diagonal principal de L.
    L[i,i] <- 1

    for (j in (i+1):n_incognitas) {
      L[i,j] <- 0
    }
  }

  #El último lo completo a mano
  L[n_incognitas, n_incognitas] <- 1

  U[1,1] <- matriz_coeficientes[1,1]


  if(U[1,1] == 0){
    return("factorizacion imposible")
  }
```

```r
# -------- PASO 2

for (j in 2:n_incognitas) {
  U[1,j] <- matriz_coeficientes[1,j]
  L[j,1] <- matriz_coeficientes[j,1]/U[1,1]
}


# -------- PASO 3

for(i in 2:(n_incognitas-1)){

  # -------- PASO 4
  suma <- 0
  for (k in 1:(i-1)) {
    suma <- suma + L[i,k]*U[k,i]
  }

  U[i,i] <- matriz_coeficientes[i,i] - suma


  if(U[i,i] == 0){
    return("factorizacion imposible")
  }

  # -------- PASO 5

  for (j in (i+1):n_incognitas) {
    sumaU <- 0
    sumaL <- 0
    for (k in 1:(i-1)) {
      sumaU <- sumaU + L[i,k]*U[k,j]
      sumaL <- sumaL + L[j,k]*U[k,i]
    }

    U[i,j] <- (1/L[i,i])* (matriz_coeficientes[i,j] - sumaU)
    L[j,i] <- (1/U[i,i])* (matriz_coeficientes[j,i] - sumaL)

  }


}


# -------- PASO 6


suma <- 0
for (k in 1:(n_incognitas-1)) {
  suma <- suma + L[n_incognitas,k]*U[k,n_incognitas]
}
U[n_incognitas, n_incognitas] <- matriz_coeficientes[n_incognitas,n_incognitas] - suma
```

```r
  # -------- PASO 7
  return(list("L" = L, "U" = U))
}

A <- matrix(c(1, 1, 0, 3,
              2, 1, -1, 1,
              3, -1, -1, 2,
              -1, 2, 3, -1), nrow = 4, ncol = 4, byrow = TRUE)

b <- matrix(c(1, 1, -3, 4), nrow = 4, ncol = 1, byrow = TRUE)

LU(matriz_coeficientes = A)
```

```
## $L
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    2    1    0    0
## [3,]    3    4    1    0
## [4,]   -1   -3    0    1
##
## $U
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    0    3
## [2,]    0   -1   -1   -5
## [3,]    0    0    3   13
## [4,]    0    0    0  -13
```

```r
L <- LU(A)$L
U <- LU(A)$U

L%*%U
```

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