Cholesky

Uriel Paluch

14/9/2021

Se usa una idea similar al algoritmo LU pero con una única matriz L.

La matriz L que se obtiene mediante la descomposición de CHolesky tiene aplicaciones en Simulación de Monte Carlo de variables aleatorias multivariadas.
- La matriz A que se factoriza n estas aplicaciones es la matriz de covarianzas, o bien la matid de correlaciones
- La matriz de covarianzas (y de correlaciones), deben ser semi-definidas positivas, pero en la mayoría de las aplicaciones son definidas positivas

Una matriz A es definida positiva si es simétrica y si $x^t Ax > 0$ para cada vector n-dimensional de $x \neq 0$. Cuando se hace este calculo se obtiene una matriz de 1X1, puesto que estamos trabajando con variables y no con incognitas va a quedar una ecuación. En el libro hay un ejemplo en el cual el resultado es las x elevadas al cuadrado, por lo tanto, es definida postiva (> 0) a menos que todos los valores de x sean 0.

Teorema 6.24:

Si A es una matriz de n X n una matriz es definida positiva, entonces:
i. A tiene inversa
ii. $a_{ii} > 0$ para cada $i = 1, 2, ..., n$; iii. $max_{1 \leq k,j \leq n} |a_{kj}| \leq max_{q \leq i \leq n} |a_{ii}|$
iv. $(a_{ij})^2 \leq a_{ii} a_{jj}$ para cada $i \neq j$

Corolario 6.28:

Una matriz A es definida positiva si y sólo si A se puede factorizar en la forma LL^t, donde L es triangular inferior con entradas diagonales diferentes a cero.

CREO que los elementos diagonales deben ser $\neq$ de cero.

```r
Cholesky <- function(A){
  n <- nrow(A)

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

  # Paso 1 ------------------------------------------------------------------
  L[1,1] <- sqrt(A[1,1])


  # Paso 2 ------------------------------------------------------------------
  for (j in 2:n){
    L[j,1] <- A[j,1]/L[1,1]
  }
```

```
  # Paso 3 ---------------------------------------------------------------
  for (i in 2:(n-1)) {

    # Paso 4 ---------------------------------------------------------------
    suma <- 0
    for (k in 1:(i-1)) {
      suma <- suma + L[i,k]^2
    }

    L[i,i] <- sqrt(A[i,i] - suma)


    # Paso 5 ---------------------------------------------------------------
    for (j in (i+1):n) {

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

      L[j,i] <- (A[j,i] - suma)/L[i,i]

    }
  }

  # Paso 6 ---------------------------------------------------------------
  suma <- 0
  for (k in 1:(n-1)) {
    suma <- suma + L[n,k]^2
  }

  L[n,n] <- sqrt(A[n,n]-suma)

  return(L)

}

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

test <- Cholesky(A)

print(test)
```

```
##            [,1]       [,2]     [,3]
## [1,]  1.4142136  0.0000000 0.000000
## [2,] -0.7071068  1.2247449 0.000000
## [3,]  0.0000000 -0.8164966 1.154701
```

```
print(test%*%t(test))
```

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

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

Ejercicios de la guía:

A:

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

test <- Cholesky(A)

print(test)
```

```
##            [,1]       [,2]     [,3]
## [1,]  1.4142136  0.0000000 0.000000
## [2,] -0.7071068  1.2247449 0.000000
## [3,]  0.0000000 -0.8164966 1.154701
```

B:

```r
B <- matrix(c(6, 2, 1, -1,
              2, 4, 1, 0,
              1, 1, 4, -1,
              -1, 0, -1, 3), nrow = 4, ncol = 4, byrow = TRUE)

test <- Cholesky(B)

print(test)
```

```
##            [,1]      [,2]       [,3]     [,4]
## [1,]  2.4494897 0.0000000  0.0000000 0.000000
## [2,]  0.8164966 1.8257419  0.0000000 0.000000
## [3,]  0.4082483 0.3651484  1.9235384 0.000000
## [4,] -0.4082483 0.1825742 -0.4678877 1.606574
```

C:

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

test <- Cholesky(C)

print(test)
```

```
##      [,1]       [,2]      [,3]     [,4]
## [1,]  2.0  0.0000000 0.0000000 0.000000
## [2,]  0.5  1.6583124 0.0000000 0.000000
## [3,]  0.5 -0.7537784 1.0871146 0.000000
## [4,]  0.5  0.4522670 0.0836242 1.240347
```

D:

```r
D <- matrix(c(1, 2, 4, 7,
              2, 13, 23, 38,
              4, 23, 77, 122,
              7, 38, 122, 294), nrow = 4, ncol = 4, byrow = TRUE)

test <- Cholesky(D)

print(test)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    2    3    0    0
## [3,]    4    5    6    0
## [4,]    7    8    9   10
```