3. Factorizacion de Cholesky

En estadística a menudo aparecen sistemas de ecuaciones Ax=b donde la matriz A es definida positiva. Por ejemplo, el sistema de ecuaciones normales de un modelo de regresión lineal tiene la forma X'Xb=y. Aquí A=X'X es definida positiva y simétrica. Se podría usar Eliminación Gaussiana para resolver el sistema de ecuaciones, pero no se explotaría la propiedad de definida positiva.
Si la matriz A es simétrica y definida positiva en lugar de factorizarse como LU, puede ser factorizada como A=HH', donde H es una matriz triangular inferior, esta es llamada la Factorización Cholesky.
La existencia de la factorización Cholesky puede ser mostrada a traves de la factorización LU. Es decir, A=LU. La matriz triangular superior U puede ser escrita como $U=DU_1$ donde $D=diag(u_{11},u_{22},\ldots,u_{nn})$ y U_1 es una matriz triangular superior con unos en su diagonal. Luego, $A=LDU_1$ y como A=A' se tiene que $LDU_1=U'_1DL'$ de donde $D=(U'_1)^{-1}LDU_1(L')^{-1}$
La matriz $(U'_1)^{-1}L$ es una matriz triangular inferior unitaria y la matriz $U_1(L')^{-1}$ es una matriz triangular superior unitaria. Por lo tanto,

$$(U'_1)^{-1}L= U_1(L')^{-1} =I$$

En consecuencia, $U_1=L'$ y A puede ser escrita como A=LDL' donde L es una matriz triangular inferior unitaria. Cuando A es definida positiva los elementos de la diagonal de D son positivas y se puede escribir $D=D^{1/2}D^{1/2}$, donde $D^{1/2}=diag((\sqrt{u_{11}},\sqrt{u_{22}},\ldots.\sqrt{u_{nn}})$
Así que.

$$A=LDL'=LD^{1/2}D^{1/2}L'=HH', \text{ con } H=LD^{1/2}.$$

Es decir,

$$\begin{bmatrix} a_{11} & a_{12} & \ldots. & a_{1n} \\ a_{21} & a_{22} & \ldots.. & a_{2n} \\ \ldots.. & \ldots.. & \ldots... & \ldots.. \\ a_{n1} & a_{n2} & \ldots.. & a_{nn} \end{bmatrix} = \begin{bmatrix} h_{11} & 0 & \ldots. & 0 \\ h_{21} & h_{22} & \ldots.. & 0 \\ \ldots.. & \ldots.. & \ldots... & \ldots.. \\ h_{n1} & h_{n2} & \ldots.. & h_{nn} \end{bmatrix} \begin{bmatrix} h_{11} & h_{21} & \ldots. & h_{n1} \\ 0 & h_{22} & \ldots.. & h_{n2} \\ \ldots.. & \ldots.. & \ldots... & \ldots.. \\ 0 & 0 & \ldots.. & h_{nn} \end{bmatrix}$$

De donde

$$h_{11} = \sqrt{a_{11}} \qquad h_{i1} = \frac{a_{i1}}{h_{11}} \qquad i=1,2\ldots n$$

$$\sum_{k=1}^{i} h_{ik}^2 = a_{ii}, \qquad a_{ij} = \sum_{k=1}^{j} h_{ik} h_{jk} \qquad j<i$$

3.1 Algoritmo para la Factorización Cholesky.

Dada una matriz simétrica y definida positiva A de orden n x n, el siguiente algoritmo calcula la matriz H de la factorización Cholesky. H es calculada fila por fila y es almacenada en la parte triangular inferior de A.

For k=1,2…..n do
 For i=1,2,….,k-1 do

$$a_{ki}=h_{ki}=\frac{1}{h_{ii}}\left(a_{ki}-\sum_{j=1}^{i-1}h_{ij}h_{kj}\right)$$

$$a_{kk}=h_{kk}=\sqrt{a_{kk}-\sum_{j=1}^{k-1}h_{kj}^2}$$

Nota: Cuando k=1 el loop interior no es calculado. Al ser A definida positiva se asegura que la cantidad dentro de la raíz cuadrada es siempre positiva.

Ejemplo 3. Obtener la factorización Cholesky de la matriz

$$A=\begin{bmatrix} 4 & 2 & 2 & 4 \\ 2 & 5 & 7 & 0 \\ 2 & 7 & 19 & 11 \\ 4 & 0 & 11 & 25 \end{bmatrix}$$

A) Manualmente
B) Usando la función **Chol** de R y MATLAB

Solución:

a) Cálculo de la primera fila de H (k=1)

$h_{11}=\sqrt{4}=2$

Cálculo de la Segunda fila de H (k=2)

$h_{21}=\frac{a_{21}}{h_{11}}=\frac{2}{2}=1$

$h_{22}=\sqrt{a_{22}-h_{21}^2}=\sqrt{5-1^2}=2$

Cálculo de la tercera fila de H (k=3)

$h_{31}=\dfrac{a_{31}}{h_{11}}=\dfrac{2}{2}=1$

$h_{32}=\dfrac{1}{h_{22}}(a_{32}-h_{21}h_{31})=\dfrac{1}{2}(7-1)=3$

$h_{33}=\sqrt{a_{33}-(h_{31}^2+h_{32}^2)}=\sqrt{19-1^2-3^2}=3$

Cálculo de la cuarta fila de H (k=4)

$h_{41}=\dfrac{a_{41}}{h_{11}}=\dfrac{4}{2}=2$

$h_{42}=\dfrac{1}{h_{22}}(a_{42}-h_{21}h_{41})=\dfrac{1}{2}(0-2)=-1$

$h_{43}=\dfrac{1}{h_{33}}(a_{43}-h_{31}h_{41}-h_{32}h_{42})=\dfrac{1}{3}(11-1*2-3*(-1))=4$

$h_{44}=\sqrt{a_{44}-(h_{41}^2+h_{42}^2+h_{43}^2)}=\sqrt{25-2^2-(-1)^2-4^2}=2$

b) En R,

```
> A=cbind(c(4,2,2,4),c(2,5,7,0),c(2,7,19,11),c(4,0,11,25))
> A
     [,1] [,2] [,3] [,4]
[1,]    4    2    2    4
[2,]    2    5    7    0
[3,]    2    7   19   11
[4,]    4    0   11   25
> chol(A)
     [,1] [,2] [,3] [,4]
[1,]    2    1    1    2
[2,]    0    2    3   -1
[3,]    0    0    3    4
[4,]    0    0    0    2

> t(chol(A))%*%chol(A)
     [,1] [,2] [,3] [,4]
[1,]    4    2    2    4
[2,]    2    5    7    0
[3,]    2    7   19   11
[4,]    4    0   11   25
>
```

Usando MATLAB

```
» A

A =

     4     2     2     4
     2     5     7     0
     2     7    19    11
     4     0    11    25

» L=chol(A)

L =

     2     1     1     2
     0     2     3    -1
     0     0     3     4
     0     0     0     2

»
```

Notar que tanto R como MATLAB dan la matriz triangular superior.

3.2 Solución del sistema de ecuaciones Ax=b usando la factorización Cholesky

El sistema de ecuaciones Ax=b se transforma en HH'x=b. Luego, se resuelve el sistema triangular inferior Hy=b y despues el sistema triangular superior H'x=y.

Ejemplo 4 : Resolver el sistema Ax=b, donde A está dado en el ejemplo 3 y el vector de términos independientes b es (-1 1 5/2 1/4)'.

Solución: Resolviendo el sistema triangular inferior Hy=b, para y con

$$H=\begin{matrix} 2 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 1 & 3 & 3 & 0 \\ 2 & -1 & 4 & 2 \end{matrix}$$

se obtiene y_1=-1/2, y_2=1/2(1+1/2)=3/4, y_3=1/3(5/2-9/4 +1/2)=1/4, y y_4=1/2(1/4-1+3/4+1)=1/2. Resolviendo el sistema H'x=y para x , con

$$H' = \begin{matrix} 2 & 1 & 1 & 2 \\ 0 & 2 & 3 & -1 \\ 0 & 0 & 3 & 4 \\ 0 & 0 & 0 & 2 \end{matrix}$$

se obtiene, x_4=1/4, x_3=1/3(1/4-1)=-1/4, x_2=1/2(3/4+1/4+3/4)=7/8 y, x_1=1/2(-1/2-1/2+1/4-7/8)=-13/16.

3.3 Aplicación a Regresión (Cuadrados Mínimos)

Dado el conjunto de datos ($\mathbf{x}_i$,y_i) (i=1,….,n) donde $\mathbf{x}_i$ es un vector de dimension p de mediciones que se han hecho en la i-ésima observación y y_i es la respuesta que se ha observado, x es llamado el **vector de variables predictoras** y y es la **variable de respuesta.** Por ejemplo, se ha observado que una casa que cuesta 250,000 tiene 600 metros cuadrados de area construida, 6 habitaciones y 25 años de antiguedad. Es decir, se tiene el dato ((600,6,25),250,000) Supongamos que tenemos como 150 de estos datos.
El objetivo en regresión lineal es construir un modelo (lineal) que describa el comportamiento de la variable de respuesta según el comportamiento de las variables predictoras. En nuestro caso sería algo como:

Precio=b_o+b_1*area+b_2*habitaciones+b_3*antiguedad +error aleatorio

Los coeficientes b_o, b_1, b_2 y b_3 deben ser estimados usando los datos tomados, y el proceso es conocido como **ajustar un modelo de regresión lineal multiple**.

El modelo aplicado a todas las observaciones puede ser escrito matricialmente como

Y-XB+e,

donde Y es el vector de variable de respuesta,
X es una matriz n x p+1, considerando que hemos incluido el intercepto, es decir cada fila de X es de la forma (1,$\mathbf{x}_i$),
B es un vector de dimensión p+1 de parámetros a ser estimados,
e es un vector de errores aleatorios.

La idea (K. Gauss, 1828) para estimar B es minimizar la suma de cuadrados de los errores Q(B)=$||e||^2$=e'e=(Y-XB)'(Y-XB) con respecto a B.

Derivando con respecto a B e igualando a cero se obtiene el sistema de ecuaciones lineales

$$X'XB=X'Y$$

éste es llamado las **ecuaciones normales del modelo.**
X'X es claramente simétrica. Por otro lado, si X es de rango completo, en nuestro caso rango(X)=p+1, entonces $w\neq 0$ implica que $Xw\neq 0$. En consecuencia, $w'X'Xw=\|Xw\|^2>0$ para todo $w\neq 0$, así que X'X también es definida positiva y la Factorización Cholesky puede ser aplicada. Hay que notar que si X no es de rango completo entonces la propiedad de definida positiva no se cumple.
La existencia y unicidad de la solución $\hat{B}$ de las ecuaciones normales está condicionada a que X sea de rango completo. La forma explícita de la solución es

$$\hat{B}=(X'X)^{-1}X'Y$$

Otras cantidades de interés son:
El vector de valores ajustados o predichos:

$$\hat{Y}=X\hat{B}=X(X'X)^{-1}X'Y$$

El vector de residuales:

$$\hat{e}=Y-\hat{Y}$$

La suma de cuadrados del error o residual

$$\text{SSE}=e'e=(Y-X\hat{B})'(Y-X\hat{B})=Y'[I-X(X'X)^{-1}X']Y$$

La matriz de Covarianza del vector de parametros estimados $\hat{B}$

$$(X'X)^{-1}\sigma^2$$

donde σ^2 es la varianza poblacional de las y_i (o equivalente de los errores e_i). Esta varianza es estimada por MSE=$\frac{SSE}{n-p-1}$ donde p es el número de variables predictoras.
A $(X'X)^{-1}$ también le dicen la matriz no escalada de covaianza.

Algoritmo para regresión usando Factorización Cholesky

El siguiente algoritmo resuelve las ecuaciones normales del modelo de regresión y calcula ciertas cantidades importantes

a) Calcular w=X'Y
b) Calcular la factorizacion Cholesky HH' de X'X.

c) Resolver el sistema triangular Hz=w para z
d) Resolver el sistema triangular $H'\hat{B} = z$ para $\hat{B}$
e) Calcular la suma de cuadrados residual por SSE=Y'Y-z'z
f) Calcular el vector de predicciones $\hat{Y} = X\hat{B}$
g) Invertir H
h) Calcular la matriz de covarianza no escalada, usando $(X'X)^{-1}=H^{-1}{}'H^{-1}$

Ejemplo 5. Ajustar un modelo de regresión lineal multiple para el siguiente conjunto de datos

Y	x1	x2	x3
12	4	5	6
23	8	9	12
41	12	3	23
89	11	17	21
18	8	19	39
39	19	21	45
56	9	22	42
9	13	9	18

Usando R y MATLAB.
Solución:
En R,
Haremos uso de las funciones forwardsolve y backsolve

```
> eje1=read.table("c://computacional//ejechol.txt",header=T)
> eje1
  Y x1 x2 x3
1 12  4  5  6
2 23  8  9 12
3 41 12  3 23
4 89 11 17 21
5 18  8 19 39
6 39 19 21 45
7 56  9 22 42
8  9 13  9 18
> X=cbind(rep(1,8),eje1[,2:4])
> X
  rep(1, 8) x1 x2 x3
1         1  4  5  6
2         1  8  9 12
3         1 12  3 23
4         1 11 17 21
5         1  8 19 39
6         1 19 21 45
```

```
7        1  9 22 42
8        1 13  9 18
> dim(X)
[1] 8 4
> X=as.matrix(X)
> colnames(X)=NULL
> X
  [,1] [,2] [,3] [,4]
1    1    4    5    6
2    1    8    9   12
3    1   12    3   23
4    1   11   17   21
5    1    8   19   39
6    1   19   21   45
7    1    9   22   42
8    1   13    9   18
> y=eje1[,1]
> y
[1] 12 23 41 89 18 39 56  9
> w=t(X)%*%y
> H=chol(t(X)%*%X)
> H
         [,1]     [,2]      [,3]     [,4]
[1,] 2.828427 29.69848 37.123106 72.83200
[2,] 0.000000 11.74734  6.682364 20.68553
[3,] 0.000000  0.00000 18.660681 26.47394
[4,] 0.000000  0.00000  0.000000 18.72803
> z=forwardsolve(t(H),w)
> z
          [,1]
[1,] 101.46982
[2,]  16.64207
[3,]  26.73625
[4,] -16.62288
> b=backsolve(H,z)
> #calculo del vector de coeficientes de regresion
> b
           [,1]
[1,]  8.1911264
[2,]  1.4482906
[3,]  2.6919893
[4,] -0.8875937
> #calculo de la suma de cuadrados del error
> sse=t(y)%*%y-t(z)%*%z
> sse
         [,1]
```

```
[1,] 3772.769
> #calculo del vector de predicciones
> yhat=X%*%b
> yhat
     [,1]
1 22.11867
2 33.35423
3 13.23193
4 51.24667
5 36.30909
6 52.29871
7 43.17057
8 35.27012
> #calculo de la matriz de covarianza no escalonada
> cov=t(solve(H))%*%solve(H)
> cov
            [,1]       [,2]        [,3]        [,4]
[1,]  0.12500000 -0.3160129 -0.13550820  0.05448148
[2,] -0.31601286  0.8061594  0.33998376 -0.14207039
[3,] -0.13550820  0.3399838  0.15070076 -0.06156841
[4,]  0.05448148 -0.1420704 -0.06156841  0.03492952

>#Verificacion de resultados
>#Haciendo la regresion lineal multiple
>l1=lm(Y~.,data=eje1)
> attributes(l1)
$names
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"

$class
[1] "lm"

> # Mostrando el vector de coeficientes de regresion
> l1$coef
(Intercept)          x1          x2          x3
  8.1911264   1.4482906   2.6919893  -0.8875937
> #Mostrando los valores estimados(predichos) de la varaiable de respuesta
 > l1$fitted.values
       1        2        3        4        5        6        7        8
22.11867 33.35423 13.23193 51.24667 36.30909 52.29871 43.17057 35.27012
>
```

En Matlab
Haremos uso de las funciones forelem y backsub que están disponibles en el folder c:\matlab\acuna

```
» X=[1 4 5 6; 1 8 9 12; 1 12 3 23; 1 11 17 21; 1 8 19 39; 1 19 21 45;1 9 22 42; 1 13 9 18]

» X
X =

     1     4     5     6
     1     8     9    12
     1    12     3    23
     1    11    17    21
     1     8    19    39
     1    19    21    45
     1     9    22    42
     1    13     9    18

» y=[12;23;41;89;18;39;56;9]
» y

y =

    12
    23
    41
    89
    18
    39
    56
     9

» w=X'*y

w =

         287
        3209
        4377
        8131

» H=chol(X'*X)

H =
```

```
    2.8284   29.6985   37.1231   72.8320
         0   11.7473    6.6824   20.6855
         0         0   18.6607   26.4739
         0         0         0   18.7280

» addpath c:\matlab\acuna
» z=forelm(H',w)

z =

  101.4698
   16.6421
   26.7363
  -16.6229

» b=backsub(H,z)

b =

    8.1911
    1.4483
    2.6920
   -0.8876


 » load y
» y

y =

    12
    23
    41
    89
    18
    39
    56
     9

» load z
» z

z =

  101.4698
   16.6421
```

```
  26.7363
 -16.6229

» %calculo de la suma de cuadrados del error
» sse=y'*y-z'*z

sse =

  3.7728e+003

» %Calculo del vector de predicciones

» yhat=X*b

yhat =

   22.1187
   33.3542
   13.2319
   51.2467
   36.3091
   52.2987
   43.1706
   35.2701

» %Calculo de la matriz de covarianza no escalada

» cov=(inv(H))'*inv(H)

cov =

    0.1250   -0.3160   -0.1355    0.0545
   -0.3160    0.8062    0.3400   -0.1421
   -0.1355    0.3400    0.1507   -0.0616
    0.0545   -0.1421   -0.0616    0.0349

»
```

Ahora, solamente para comparar resultados usaremos la función **regress** del toolbox **stats** de MATLAB. Esta función no usa la factorización Cholesky, si no la factorización QR que sera vista mas adelante.

```
» load X
» load y

» addpath c:\matlab\toolbox\stats
```

```
» help regress

 REGRESS Multiple linear regression using least squares.
    b = REGRESS(y,X) returns the vector of regression coefficients, B.
    Given the linear model: y = Xb,
    (X is an nxp matrix, y is the nx1 vector of observations.)
    [B,BINT,R,RINT,STATS] = REGRESS(y,X,alpha) uses the input, ALPHA
    to calculate 100(1 - ALPHA) confidence intervals for B and the
    residual vector, R, in BINT and RINT respectively.
    The vector STATS contains the R-square statistic along with the F
    and p values for the regression.



» b=REGRESS(y,X)

b =

    8.1911
    1.4483
    2.6920
   -0.8876

» [b,bint,r,rint,stats]=REGRESS(y,X,.05)

b =

    8.1911
    1.4483
    2.6920
   -0.8876


bint =

  -81.0179   97.4001
   -7.4006   10.2972
   -5.2202   10.6042
   -5.4406    3.6654


r =

  -10.1187
  -10.3542
   27.7681
```

```
   37.7533
  -18.3091
  -13.2987
   12.8294
  -26.2701


rint =

  -77.8872   57.6499
  -92.3572   71.6487
   16.7447   38.7915
    2.1904   73.3163
  -83.3236   46.7054
  -65.2154   38.6179
  -54.6301   80.2890
  -96.3175   43.7773


stats =

    0.2516    0.4482    0.7322
```