

Aproximación discreta de mínimos cuadrados con LAPACK

Pablo Santamaría

v0.2 (Julio 2009)

1. Planteo del problema

En general, los problemas que aparecen en la ciencia nos enfrentan a la observación de cantidades que cambian en el tiempo y/o el espacio. Supongamos que este cambio puede ser modelado por una relación matemática $y(x) = f(c_1, \dots, c_n; x)$ donde x es el parámetro que describe el cambio y c_1, \dots, c_n son n cantidades desconocidas, llamadas *parámetros del modelo*, cuyos valores queremos determinar. En particular consideremos el caso en que el modelo es *lineal* en sus parámetros, esto es, la relación funcional f es lineal respecto de los parámetros c_1, \dots, c_n y por lo tanto puede expresarse como

$$f(c_1, \dots, c_n; x) = \sum_{j=1}^n c_j \phi_j(x),$$

para n funciones $\phi_j(x)$ del parámetro x . Un conjunto de m observaciones de f proveerá de valores medidos y_i afectados de errores ϵ_i :

$$\begin{aligned} y_i &= y(x_i) + \epsilon_i \\ &= \sum_{j=1}^n \phi_j(x_i) c_j + \epsilon_i, \quad i = 1, 2, \dots, m. \end{aligned}$$

Definiendo la *matriz de diseño*, $m \times n$, A de elementos $a_{ij} = \phi_j(x_i)$, el *vector de parámetros*, $n \times 1$, \mathbf{x} de elementos c_i , el *vector de medidas*, $m \times 1$, \mathbf{b} de elementos y_i , el *vector de residuos*, $m \times 1$, \mathbf{r} de elementos ϵ_i , el conjunto de ecuaciones anteriores puede escribirse matricialmente como

$$A \mathbf{x} + \mathbf{r} = \mathbf{b}.$$

Esta relación constituye nuestro *modelo lineal* de los datos. Si $m \geq n$ (más observaciones que parámetros) se trata ahora determinar el conjunto de parámetros \mathbf{x} que satisfaga mejor, en algún sentido, el modelo lineal. De acuerdo al *método de mínimos cuadrados*, los estimadores de mínimos cuadrados de los parámetros son aquellos valores que minimizan la norma euclídea del vector de residuos:

$$\|\mathbf{r}\|_2 = \|\mathbf{b} - A \mathbf{x}\|_2 = (\mathbf{r}^t \mathbf{r})^{1/2}.$$

Este requisito conduce a que la solución de mínimos cuadrados debe satisfacer las *ecuaciones normales*

$$(A^t A) \mathbf{x} = A^t \mathbf{b},$$

de donde se sigue que, cuando $A^t A$ es no-singular, la solución de mínimos cuadrados existe y es única. La condición *necesaria y suficiente* para la existencia de una solución única es que las columnas de la matriz A sean linealmente independientes o, dicho en forma equivalente, que el rango de la matriz A sea n . En tal caso, es fácil ver que la matriz $A^t A$ es simétrica y definida positiva¹, con lo cual un método numérico apropiado para resolver las ecuaciones normales es el

¹Esto es, $\mathbf{x}^t A \mathbf{x} > 0$ para todo $\mathbf{x} \neq 0$.

método de Choleski. Sin embargo, en la práctica, muy a menudo las columnas de A son aproximadamente linealmente dependientes lo cual conduce a un sistema de ecuaciones normales con una matriz mal condicionada. Por tal motivo resulta fundamental recurrir a otra forma de resolución numérica del problema. En particular el *método QR* evita la formación de las ecuaciones normales y garantizan la estabilidad de la solución frente a los errores de redondeo. Este método se basa en la existencia de la factorización QR de la matriz A $m \times n$, cuando la misma es de rango n ,

$$A = QR,$$

siendo Q es una matriz $m \times n$ cuyas columnas son una base ortonormal del espacio columna de A (con lo cual $Q^t Q = I_n$) y R es una matriz cuadrada de orden n triangular superior con elementos positivos sobre la diagonal. En tal caso las ecuaciones normales toman la forma

$$(QR)^t(QR) \mathbf{x} = (QR)^t \mathbf{b},$$

es decir

$$R^t R \mathbf{x} = R^t Q^t \mathbf{b}$$

y como R es no-singular, se obtiene $R \mathbf{x} = Q^t \mathbf{b}$. Así, conocida la factorización QR de A , la solución de mínimos cuadrados resulta, pues, por sustitución hacia atrás del sistema triangular superior

$$R \mathbf{x} = \hat{\mathbf{b}}, \quad \hat{\mathbf{b}} = Q^t \mathbf{b}.$$

La teoría anterior puede ser aplicada en particular al caso particular, pero importante, en que la relación funcional del modelo lineal sea un polinomio de grado a lo más $(n - 1)$:

$$f(c_1, \dots, c_n; x) = \sum_{j=1}^n c_j x^{j-1} = c_1 + c_2 x + \dots + c_n x^{n-1}.$$

En este caso los parámetros a determinar son los coeficientes c_i del polinomio y la matriz de diseño \mathbf{A} tiene por elementos $a_{ij} = x_i^{j-1}$. Los estimadores de mínimos cuadrados conducen así al *ajuste de los datos por un polinomio en el sentido de mínimos cuadrados*.

2. Subrutina DGELS de la biblioteca LAPACK

El problema discreto de mínimos cuadrados queda entonces planteado como sigue: dados el vector de medidas \mathbf{b} , de dimensión m , y la matriz de diseño A , de dimensiones $m \times n$, queremos determinar el vector de estimadores \mathbf{x} , de dimensión n , que hacen mínima la norma euclídea del vector de residuos \mathbf{r} , de dimensión m , satisfaciéndose la ecuación

$$A \mathbf{x} + \mathbf{r} = \mathbf{b}.$$

Con más generalidad, podemos considerar l problemas de mínimos cuadrados:

$$A \mathbf{x}_1 + \mathbf{r}_1 = \mathbf{b}_1, \quad A \mathbf{x}_2 + \mathbf{r}_2 = \mathbf{b}_2, \quad \dots \quad A \mathbf{x}_l + \mathbf{r}_l = \mathbf{b}_l,$$

para la misma matriz de diseño A pero un conjunto distinto de \mathbf{b}_j , $j = 1, \dots, l$, vectores de medidas. Definiendo la matriz $B = [\mathbf{b}_1 | \mathbf{b}_2 | \dots | \mathbf{b}_l]$, de dimensiones $m \times l$, y la matriz de estimadores $X = [\mathbf{x}_1 | \mathbf{x}_2 | \dots | \mathbf{x}_l]$, de dimensiones $n \times l$, los sistemas de ecuaciones anteriores pueden escribirse matricialmente en la forma

$$A X + R = B,$$

siendo las columnas de $R = [\mathbf{r}_1 | \mathbf{r}_2 | \dots | \mathbf{r}_l]$, la matriz de residuos de dimensiones $m \times n$, de norma euclídea mínima para los respectivos estimadores de mínimos cuadrados².

Para datos reales de *doble precisión*, la biblioteca de subrutinas LAPACK permite resolver este conjunto de problemas de mínimos cuadrados a través de la subrutina general DGELS a través de la factorización QR . La interface o prototipo de esta subrutina es como sigue:

²Es decir, el problema se resuelve para cada \mathbf{b}_j en forma independiente, lo cual *no* es lo mismo que encontrar una matrix X que minimice $\|R\|_2 = \|B - AX\|_2$.

```

SUBROUTINE DGELS( TRANS, M, N, NRHS, A, LDA, B, LDB, WORK, LWORK, INFO )
  CHARACTER          TRANS
  INTEGER            INFO, LDA, LDB, LWORK, M, N, NRHS
  DOUBLE PRECISION  A( LDA, * ), B( LDB, * ), WORK( * )

```

donde, para nuestro problema:

TRANS: Es un dato de entrada de tipo CHARACTER de longitud uno, que debe ser asignada a 'N' en la llamada de la subrutina para el problema que estamos considerando³.

M: Es un dato de entrada de tipo entero que indica el número m de filas de la matriz de diseño A .

N: Es un dato de entrada de tipo entero que indica el número n de columnas de la matriz de diseño A .

NRHS: Es un dato de entrada de tipo entero que indica el número l de columnas de las matrices B y X .

A: Es un arreglo bidimensional de datos de doble precisión que como entrada contiene a la matriz de diseño A y, como salida, los detalles de la factorización QR realizada en la resolución del problema.

LDA: Es un dato de entrada de tipo entero que indica el número máximo de filas con que ha sido declarado el arreglo bidimensional que se utiliza para almacenar la matriz de diseño A . Típicamente, este arreglo será declarado en el programa principal con dimensiones físicas suficientemente grandes para los problemas a considerar en la forma:

```

INTEGER MMAX          ! Número máximo de parametros
PARAMETER (MMAX = 50)
INTEGER MMAX          ! Número máximo de observaciones
PARAMETER (MMAX = 100)
DOUBLE PRECISION A(MMAX,MMAX)

```

Entonces LDA = MMAX.

B: Es un arreglo bidimensional de datos de doble precisión que como entrada contiene la matriz B . Como salida, si INFO = 0, el arreglo es sobrescrito con la solución de mínimos cuadrados como sigue: Para cada columna j de B las filas de 1 a N contienen al vector de parámetros de mínimos cuadrados \mathbf{x}_j , en tanto que la suma de los cuadrados de los elementos de las filas de N+1 a M nos da $\|\mathbf{r}_j\|_2^2$, esto es, el cuadrado de la norma euclídea del vector residuo correspondiente.

LDB: Es un dato de entrada de tipo entero que indica el número máximo de filas con que ha sido declarado el arreglo bidimensional que se utiliza para almacenar la matriz B . Típicamente, este arreglo será declarado en el programa principal con dimensiones físicas suficientemente grandes para los problemas a considerar en la forma:

```

INTEGER MMAX          ! Número máximo de observaciones
PARAMETER (MMAX = 100)
INTEGER LMAX          ! Número máximo de problemas de mínimos cuadrados
PARAMETER (LMAX = 10)
DOUBLE PRECISION B(MMAX,LMAX)

```

Entonces LDB = MMAX.

³La asignación a 'N' implica que la matriz involucrada en el problema es efectivamente A y no A^t .

WORK: Es un arreglo unidimensional de datos de doble precisión utilizado para cálculos intermedios de la subrutina.

LWORK: Es un dato de entrada de tipo entero que indica la dimensión del arreglo **WORK**, cuyo valor debe ser al menos $MN + \max\{MN, NRHS\}$. En el programa principal, con las especificaciones anteriores para los arreglos bidimensionales que contendrán a A y B , el vector **WORK** debe ser entonces declarado como:

```
DOUBLE PRECISION WORK(MMAX*NMAX+MAX(MMAX*NMAX,LMAX))
```

INFO: Es un dato de salida de tipo entero utilizado como clave de éxito o error de la subrutina. Si el valor devuelto es igual a 0 entonces la subrutina ha calculado exitosamente las soluciones de mínimos cuadrados. Si el valor devuelto es negativo e igual a $-i$, esto significa que el argumento i -ésimo en la llamada de la subrutina ha sido asignado a un valor incorrecto. Finalmente, si el valor devuelto es positivo e igual a i , esto significa que el elemento i -ésimo de la diagonal del factor triangular de A es cero, por lo tanto la matriz A es deficiente de rango y la solución de mínimos cuadrados no puede ser calculada.

3. Ajuste de polinomios en el sentido de mínimos cuadrados

Como aplicación de la subrutina **DGELS** consideremos el problema de aproximar, en el sentido de mínimos cuadrados, un conjunto de datos $\{(x_i, y_i), i = 1, \dots, m\}$ con un polinomio $f(x) = \sum_{k=1}^n c_k x^{k-1}$ de grado a lo más $n - 1 < m$. Aquí los estimadores del modelo son los n coeficientes del polinomio de grado $g = n - 1$. El siguiente programa implementa la solución. Los datos se asumen dados en un archivo de texto plano con los valores de las abscisas y ordenadas en una línea para cada i .

```

PROGRAM AJUSTEPOLINOMIAL
C -----
C Bloque de declaración de tipo y almacenamiento
C -----
IMPLICIT NONE
INTEGER NMAX, MMAX
C -----
PARAMETER (NMAX = 50)      ! Número máximo de parametros
PARAMETER (MMAX = 100)    ! Número máximo de datos
C -----
DOUBLE PRECISION X(MMAX)
DOUBLE PRECISION Y(MMAX)
DOUBLE PRECISION A(MMAX,NMAX)
DOUBLE PRECISION WORK(2*MMAX*NMAX)
C -----
INTEGER G, N, M, I, J, KODE
C -----
CHARACTER(80) ARCHIVO
C -----
C Leer el nombre del archivo de datos
C -----
WRITE(*,'(A,$)') ' Archivo de datos? '
READ(*,*) ARCHIVO
C -----
C Bloque de lectura de datos
C -----

```

```

OPEN(8,FILE=ARCHIVO,STATUS='OLD',IOSTAT=KODE)
IF (KODE.NE.0) THEN
  WRITE(*,*) 'El archivo de datos no pudo ser leído'
  STOP
ENDIF

M = 0
KODE = 0
DO WHILE (KODE.EQ.0)
  M = M+1
  READ(8,*,IOSTAT=KODE) X(M),Y(M)
END DO
CLOSE(8)
M = M-1
WRITE(*,*) 'Número de datos leídos = ', M
C -----
C Leer el grado del polinomio
C -----
WRITE(*,'(A,$)') ' Grado del polinomio? '
READ(*,*) G
N = G + 1 ! El número de parámetros es uno mas que el
          ! grado del polinomio
C -----
C Controlar que el número de parámetros no sea mayor
C que el número de observaciones
C -----
IF(N.GT.M) THEN
  WRITE (*,'(A)')
& ' ERROR: Número de parámetros mayor que el de datos'
  STOP
ENDIF
C -----
C Bloque de procesamiento
C -----
C Construir la matriz de diseño
C -----
DO I = 1,M
  DO J = 1,N
    A(I,J) = X(I)**(J-1)
  END DO
END DO
C -----
C Resolver el problema de mínimos cuadrados con LAPACK
C -----
CALL DGELS('N',M,N,1,A,M,MAX,Y,M,WORK,2*M,MAX*N,MAX,KODE)
IF (KODE.EQ.0) THEN
C -----
C Imprimir solución
C -----
WRITE (*,*) '# GRADO DEL POLINOMIO = ', G
WRITE (*,*) '# NUMERO DE PARAMETROS = ', N
WRITE (*,*) '# PARAMETROS'
DO I=1,N
  WRITE (*,*) I, Y(I)

```

```

        END DO
        WRITE(*,*)
    ELSE
        WRITE(*,*) 'Error = ', KODE
    ENDIF
C -----
    END

```

Asumiendo que este programa está guardado en el archivo fuente `fitpol.f`, incorporamos la biblioteca de rutinas LAPACK en la compilación como sigue:

```
$ gfortran -Wall -o fitpol fitpol.f -llapack
```

Como ejemplo consideremos el conjunto de $m = 10$ datos $\{x_i, y_i\}$ dados en la siguiente tabla, la que supondremos almacenada en el archivo `datos.dat`:

x_i	y_i
-0.9	81.0
-0.7	50.0
-0.5	35.0
-0.3	27.0
-0.1	26.0
0.1	60.0
0.3	106.0
0.5	189.0
0.7	318.0
0.9	520.0

Con la elección de, por ejemplo, un polinomio cúbico, el programa arroja los siguientes resultados:

```

Archivo de datos? datos.dat
Número de datos leídos =          10
Grado del polinomio? 3
# GRADO DEL POLINOMIO =           3
# NUMERO DE PARAMETROS =          4
# PARAMETROS
   1  35.7312500000000
   2  116.501796814297
   3  319.602272727273
   4  156.347125097125

```

De este modo, considerando los coeficientes redondeados a cinco decimales, el polinomio de mínimos cuadrados de grado 3 que ajusta los datos es:

$$f(x) = 35.73125 + 116.50180x + 319.60227x^2 + 156.34713x^3$$