

Minimización por el método de Newton-Raphson

Pablo de Castro

January 11, 2015

1 Introducción

El objetivo de este trabajo es la implementación de un programa para el ajuste no lineal por mínimos cuadrados mediante minimización del χ^2 por el método de Newton-Raphson. Este programa se va a utilizar para ajustar unos datos experimentales a una función $y = \exp(ax) + bx^2$ para estimar los parámetros a y b óptimos. Adicionalmente, se van a estimar las incertumbres del ajuste a partir de la matriz de información de Fisher y simulaciones Monte Carlo.

2 Fundamento Teórico

Sea un conjunto de N medidas y_i (con errores σ_i) para valores x_i a los que se les quiere ajustar una función $f(x_i; a_j)$ que predice los valores de y_i , donde a_j con $j = 1, \dots, p$ son parámetros desconocidos. Los valores óptimos de los parámetros a_j , de acuerdo con el método de mínimos cuadrados, para el ajuste se pueden estimar minimizando el χ^2 :

$$\chi^2 = \sum_{i=1}^N \frac{[y_i - f(x_i; a_j)]^2}{\sigma_i^2}$$

que sigue aproximadamente una distribución χ^2 con $N - p$ grados de libertad, hecho que puede utilizarse para evaluar la bondad del ajuste como la probabilidad del χ^2 real sea mayor que el χ^2 observado.

La minimización para funciones sencillas se puede realizar de forma analítica, sin embargo, en general se requiere la minimización numérica del χ^2 para la obtención de los parámetros. Existen multitud de métodos de minimización numérica, aunque el método que se va utilizar en este trabajo es el Newton-Raphson (NR) que sirve para buscar el cero de la función, por lo que se puede utilizar para buscar el cero de del gradiente de χ^2 con respecto a los parámetros.

Supóngase una función $f(x_1, \dots, x_n)$ que depende de varios parámetros \mathbf{x} , se puede desarrollar en serie en primer orden alrededor de un valor inicial $\mathbf{x}^{(0)}$ según:

$$\nabla f(\mathbf{x}) = \nabla f(\mathbf{x}^{(0)}) + \mathbf{H}(\mathbf{x} - \mathbf{x}^{(0)}) + \dots$$

en donde \mathbf{x} es el vector de parámetros, $\nabla f(\mathbf{x})$ es el vector gradiente y \mathbf{H} es la matriz hessiana. Si se impone que $\nabla f(\mathbf{x}^{(0)}) = 0$, es decir que hay un extremo en $\mathbf{x}^{(0)}$, se obtiene el siguiente sistema de ecuaciones:

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \mathbf{H}^{-1} \nabla f(\mathbf{x}^{(i)})$$

que se puede resolver de forma iterativa para obtener los valores de \mathbf{x} que corresponde a un extremo. Este método puede aplicarse para encontrar el mínimo de la función χ^2 a partir de las derivadas primeras y segundas de la misma con respecto a los parámetros a_j .

Una vez que se han estimado los valores óptimos para los parámetros \hat{a}_j , la matriz de covarianza V puede obtenerse utilizando la matriz de información de Fisher, que se puede relacionar con la matriz hessiana para los mejores valores de los parámetros obtenidos. La inversa de la matriz de covarianza para el χ^2 se obtiene como:

$$(V^{-1})_{ij} = \frac{1}{2} \frac{\partial^2 \chi^2}{\partial a_i \partial a_j} \Big|_{a_j = \hat{a}_j}$$

en donde el error de los parámetros estimados se puede obtener como la raíz cuadrada de los elementos de la diagonal de la matriz de covarianza V .

Adicionalmente, en este trabajo se va a estimar el sesgo y el error en los parámetros utilizando simulaciones Monte Carlo. En las simulaciones se va suponer que los errores son gaussianos y se van a realizar un total de N simulaciones. Para cada simulación s , el valor de y_i^s puede obtenerse como:

$$y_i^s = f(x_i; a_j) + n_i^s \text{ con } s = 1, \dots, N$$

en donde n_i^s es el error simulado para cada punto de datos que es un número aleatorio que sigue una distribución gaussiana con una desviación estandar σ_i y a_j son los parámetros del modelo. El valor de los parámetros de ajuste \hat{a}_j puede volver a ser estimado para cada simulación, a partir de los cuales se puede estimar el sesgo b_j y el error $\sigma(\hat{a}_j)$:

$$b_j = \langle \hat{a}_j^s \rangle - \hat{a}_j^{data}$$

$$\sigma(\hat{a}_j) = \sqrt{\langle \hat{a}_j^2 \rangle - \langle \hat{a}_j \rangle^2}$$

en donde los paréntesis angulares indican el promedio sobre todas las simulaciones, \hat{a}_j^2 es la estimación del parámetro para cada simulación y \hat{a}_j^{data} es la estimación de los parámetros de los datos. Si el sesgo de un estimador es cero, este está insesgado. La estimación final de los parámetros se puede obtener según:

$$a_i^{final} = (\hat{a}_i - b_i) \pm \sigma(\hat{a}_i)$$

que se puede comparar con la obtenida a partir de información de Fisher.

3 Implementación

Se ha implementado un programa de Fortran, que se adjunta con este informe, para ajustar por mínimos cuadrados una función $y = \exp(ax) + bx^2$ a una serie de datos mediante el método de Newton-Raphson de acuerdo con el algoritmo descrito en el apartado anterior.

La mayor parte de la implementación se ha realizado en forma de funciones y subrutinas en el módulo *NRmin_mod* (archivo *NRmin_mod.f90*). Estos procedimientos son invocados desde el programa *NRmin* (archivo *NRmin.f90*), en el cual se pueden configurar algunos de los parámetros de ejecución como el nombre de archivo de datos, la tolerancia del ajuste, los valores iniciales de a y b o el número de simulaciones. Aunque esta forma de configuración tiene la desventaja de que es necesario compilar el programa cada vez que se cambian los parámetros, ahorra la necesidad de introducir los parámetros en cada ejecución o desarrollar un programa para la lectura de un archivo de configuración.

4 Compilación, Ejecución y Resultados

Se ha creado un archivo Makefile para automatizar la compilación de los módulos necesarios y la creación del ejecutable, una vez que se han configurado los parámetros deseados en el archivo *NRmin.f90*, cuya llamada se demuestra a continuación:

```
In [1]: !cd ../src; make
```

```
gfortran -c -O3 mod_chi2.f90
gfortran -c -O3 mod_random.f90
gfortran -c -O3 NRmin_mod.f90
gfortran -c -O3 NRmin.f90
gfortran -o NRmin.bin -O3 mod_chi2.o mod_random.o NRmin_mod.o NRmin.o
```

Una vez compilado el programa, se ha creado el archivo binario *NRmin.bin*, cuya ejecución genera el siguiente resultado para el ajuste a realizar:

```
In [2]: !cd ../src; ./NRmin.bin

--- importing data ---
Opening file: ../data/exp.dat
# data entries      25
--- minimization of \Chi^2 by Newton-Raphson ---
step   0: a =  1.00000  b =  1.00000  \Chi^2 =   41.448
step   1: a =  0.70055  b =  1.77710  \Chi^2 =   17.354
step   2: a =  0.71956  b =  1.73214  \Chi^2 =   17.253
step   3: a =  0.71976  b =  1.73195  \Chi^2 =   17.253
step   4: a =  0.71976  b =  1.73195  \Chi^2 =   17.253
convergence reached!
--- estimating covariance matrix ---
--- NR minimization results ---
a parameter: 0.71976 \pm 0.11895
b parameter: 1.73195 \pm 0.15721
--- goodness of fit ---
Chi^2 value 17.25
Degrees of freedom: 23
Probability chi^2(real)>chi^2(obtained): 0.797
--- MC simulation ---
# simulations:      100000
a parameter -> bias = -0.00264 error= 0.11788
b parameter -> bias = -0.00640 error= 0.15758
--- final results with simulation ---
a parameter: 0.72240 \pm 0.11788
b parameter: 1.73835 \pm 0.15758
--- end of the program ---
```

En este caso se han leído los datos provistos y se ha utilizado el método de Newton-Raphson con una tolerancia de 10^{-5} y unos valores iniciales $[1.0, 1, 0]$ (se han comprobado otros valores iniciales para comprobar que el no se encontraba en un mínimo relativo) para a y b . Se observa que método converge muy rápidamente, en 4 iteraciones, proporcionando unos valores de ajuste para los parámetros de:

$$a = 0.72 \pm 0.12$$

$$b = 1.73 \pm 0.16$$

las incertidumbres de los parámetros han sido en este caso calculadas a partir de la matrix de información de Fisher. El valor mínimo para el $\chi^2 = 17.25$ para $N - p = 23$ grados de libertad, lo que corresponde a una probabilidad de bondad del ajuste $p = 0.797$, que es mucho mayor que 0.01 por lo que se ha obtenido un buen ajuste.

Se han realizado un total de 100000 simulaciones, de la forma descrita en el fundamento teórico, para estimar valores precisos para el sesgo y error que se muestran en la salida de la ejecución del programa. Los valores obtenidos para el sesgo son pequeños ($b_1 \approx -0.003$ y $b_2 \approx -0.006$) pero no se puede considerar que el estimador este insesgado. Los errores estimados a partir de la simulación han sido de $\sigma(a) \approx 0.12$ y $\sigma(b) \approx 0.16$ que coinciden con estimados a partir de la matriz de Fisher por lo que se comprueba la validez de este método.

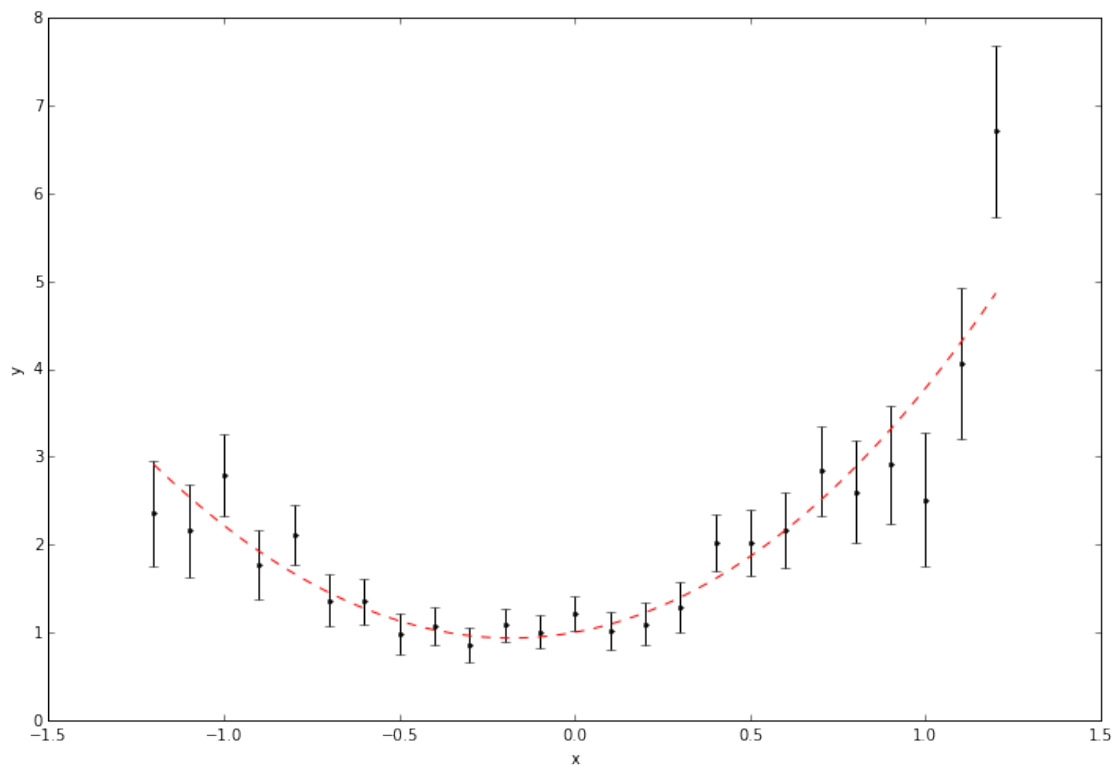
A continuación, se presentan los datos experimentales conjuntamente con el modelo a ajustar utilizando los parámetros obtenidos por el programa desarrollado. La visualización provista se ha realizado en Python, con ayuda de las bibliotecas de software NumPy y Matplotlib.

```

In [3]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
# import and plot data
dat = np.genfromtxt("../data/exp.dat")
plt.figure(figsize=(12.0,8.0))
plt.errorbar(dat[:,0],dat[:,1],dat[:,2], fmt=".k")
plt.xlabel("x")
plt.ylabel("y")
# generate model data
a = 0.72 # /pm 0.12
b = 1.73 # /pm 0.16
x = np.linspace(np.min(dat[:,0]),np.max(dat[:,0]), 1000)
plt.plot(x, np.exp(x*a)+b*x**2, "--r")

```

Out[3]: [<matplotlib.lines.Line2D at 0x1060b4510>]



Se observa, como se esperaba dado el alto valor de la bondad del ajuste, que los parámetros obtenidos ajustan bien los datos experimentales a partir del modelo propuesto.

5 Conclusión

Se ha implementado existosamente un programa en Fortran para el ajuste de una función no lineal a unos datos experimentales mediante minimización del χ^2 por método de Newton-Raphson. Las incertidumbres de los parámetros han sido obtenidas tanto a partir de la matriz de información de Fisher como de simulaciones Monte Carlo.

Se ha obtenido un valor óptimo de los parámetros para los datos provistos de $a = 0.72 \pm 0.12$ y $b = 1.73 \pm 0.16$. Los errores obtenidos mediante simulaciones coinciden con los obtenidos a partir de la matriz de Fisher.