Project: Estimation of nonlinear fitting parameters with Conjugate Gradients and Golden Section methods

Tomasz Chwiej

30th January 2025

1 Introduction

Our task is to approximate the set of n tabulated data (approximation nodes x_i, function's values y_i and weights w_i) of unknown function

$$\{(x_0, y_0, w_0),\ (x_1, y_1, w_1),\ \ldots,\ (x_{n-1}, y_{n-1}, w_{n-1})\} \tag{1}$$

with fitting function $p(x, \vec{a})$ depending on m free parameters written as elements of vector $\vec{a} = [a_0, a_1, \ldots, a_{m-1}]$, in other words, the elements of vector $\vec{a}$ need to be estimated. To solve this problem we use the mean-square approximation method, by first defining the 2-nd norm

$$f(\vec{a}) = \sum_{i=0}^{n-1} w_i \left(y_i - p(x_i, \vec{a})\right)^2 \tag{2}$$

and next minimizing the value of auxiliary non-negative objective function $f(\vec{a})$. If the objective function f depends nonlinearly on at least one variable a_i, the problem must be solved by using one of the multivariate minimization methods. In project we use Conjugate Gradients method which needs calculation of gradient $\nabla_{\vec{a}} f(\vec{a})$ in every iteration, differentation of Eq.2 provides

$$\nabla_{\vec{a}} f(\vec{a}) = \sum_{i=0}^{n-1} (-2) w_j \left(y_i - p(x_i, \vec{a})\right) \left[\frac{\partial\, p(x_i, \vec{a})}{\partial\, a_0}, \frac{\partial\, p(x_i, \vec{a})}{\partial\, a_1}, \ldots, \frac{\partial\, p(x_i, \vec{a})}{\partial\, a_{m-1}}\right] \tag{3}$$

Summation in Eq.3 reduces an explicit dependence of gradient on tabulated data, however, these data still influences on the direction of line searching in present iteration. We need only to define the fitting function $p(x, \vec{a})$.

2 Practical part

1. Assume the approximation interval spans over $x \in [x_{min}, x_{max}]$, $x_{min} = -10$ and $x_{max} = 20$. For $n = 100$ equidistant nodes distributed in this interval calculate values of "experimentally measured" function y_i and weights

$$i = 0, 1, 2, \ldots, n-1 \tag{4}$$

$$\Delta = \frac{x_{max} - x_{min}}{n-1} \tag{5}$$

$$x_i = x_{min} + \Delta \cdot i \tag{6}$$

$$y_i = p(x_i, \vec{a}) + \delta_i \tag{7}$$

$$w_i = 1.0 \tag{8}$$

where the function $p(x, \vec{a})$ will be also our aim i.e. fitting function defined below

$$p(x, \vec{a}) = a_0 \cdot \arctan(a_1 x + a_2) + a_3 \tag{9}$$

Fitting function depends on two linear parameters (a_0 and a_3) and two nonlinear (a_1 and a_2), hence the minimization problem is definitely nonlinear. Factor δ_i is an "experimental noise" and will mimic the uncertainty of single measurement

$$\delta_i = 0.1 \cdot \left(\frac{(double)rand()}{RAND\_MAX} - 0.5 \right) \tag{10}$$

Every call of function **rand()** provides different random number, it requires the **stdlib.h** (C) or **cstdlib** (C++) header.
Generate the set of data $\{(x_i, y_i, w_i)\}$ assuming the exact values of parameters a_i: $a_0 = 0.5$, $a_1 = 10$, $a_2 = -30$, $a_3 = 0.7$. Write data to file and prepare the figure showing noised data (x_i, y_i) and fitting function $p(x, \vec{a})$ which we wish to reconstruct.

2. Write the computer program implementing the Conjugate Gradient+Golden Section method (see Sec.3), calculate the gradient using Eq.3 with the following derivatives of function $p(x, \vec{a})$

$$\frac{\partial\, p(x, \vec{a})}{\partial\, a_0} = \arctan(a_1 x + a_2) \tag{11}$$

$$\frac{\partial\, p(x, \vec{a})}{\partial\, a_1} = \frac{a_0 x}{(a_1 x + a_2)^2 + 1} \tag{12}$$

$$\frac{\partial\, p(x, \vec{a})}{\partial\, a_2} = \frac{a_0}{(a_1 x + a_2)^2 + 1} \tag{13}$$

$$\frac{\partial\, p(x, \vec{a})}{\partial\, a_3} = 1 \tag{14}$$

3. Initialize the vector $\vec{a}$ with values: $a_0 = 1$, $a_1 = 1$, $a_2 = 1$, $a_3 = 1$, then iteratively find the minimum of the objective function. In calculations use the stopping criteria for Conjugate Gradients: $\|\vec{a}_{k+1} - \vec{a}_k\|_2 < 10^{-4}$, and for Golden Section: $\lambda_{l+1} - \lambda_l < 10^{-5}$, the maximum number of iterations should not exceed 1000 for CG and 100 for GS. Perform calculations for γ_i obtained with Fletcher-Reeves and Polak-Riberie formulae (see pseudocode for CG in Hints).

4. Visualize your results obtained for Fletcher-Reeves and Polak-Riberie, namely, for each case prepare a single figure showing the scattered data (x_i, y_i), exact fitting function $p(x, \vec{a}_{exact})$ and the fitting function $p(x, \vec{a})$ for the minimized vector $\vec{a}$.

5. At home prepare the report. Based on the results assess efficency of Conjugate Gradients method (for Fletcher-Reeves and Polak-Riberie γ parameter) in searching the minimum of nonlinear function. Check what will happen when the range of initial λ values in Golden Section method is extended to $\lambda_B = 1.0$ ($\lambda_A = 0$ is left unchanged) or the elements of initial vector $\vec{a}$ have much different (larger) values? Does the CG algorithm converge?

3 Computational hints

To make the main part of the numerical code more compact write separate functions calculating:

- value of fitting function

```
double p(double x, double *a)
```

- value of objective function

```
double f(int n, double *x, double *y, double *w, double *a)
```

- gradient of objective function (here it is returned in array **g**)

```
void gradient_f(int n, double *x, double *y, double *w,double *a,double *g)
```

- value of scalar product

```
double scalar_product(int n, double *x, double *y)
```

You may implement Conjugate Gradients using following code

```
input:  $\vec{a} = [1,1,1,1]$, $\varepsilon_a = 10^{-4}$, k=0, $K_{max} = 1000$
do
    $\vec{g} \leftarrow -\nabla f(\vec{a})$
    if k=0 then
        $\vec{u} \leftarrow \vec{g}$
    else
    !Fletcher-Revees
        $\gamma \leftarrow \frac{\vec{g}^T \vec{g}}{\vec{g}_{old}^T \vec{g}_{old}}$
    !or Polak-Riberie
        $\gamma \leftarrow \frac{(\vec{g} - \vec{g}_{old})^T \vec{g}}{\vec{g}_{old}^T \vec{g}_{old}}$

        $\vec{u} \leftarrow \vec{g} + \gamma \vec{u}$
    end if
!save $\vec{g}$ for next iteration
    $\vec{g}_{old} \leftarrow \vec{g}$
    $k++$
!perform univariate minimization: Golden Section
    $\lambda \leftarrow arg \min_\lambda f(\vec{a} + \lambda \vec{u})$
!save $\vec{a}_{old}$ for checking the convergence
    $\vec{a}_{old} \leftarrow \vec{a}$
!update solution
    $\vec{a} \leftarrow \vec{a} + \lambda \vec{u}$
    $s \leftarrow \|\vec{a} - \vec{a}_{old}\|_2$
while $s > \varepsilon_a$ and $k < K_{max}$
```

and the part of code calling the Golden Section methods would look like as below

```
input:  $\lambda_A = 0$, $\lambda_B = 0.1$, $r = 0.618$, l=0,$L_{max} = 100$, $\varepsilon_\lambda = 10^{-5}$
do
        l++
        $\lambda_1 \leftarrow \lambda_A + r^2 \cdot (\lambda_B - \lambda_A)$
        $\lambda_2 \leftarrow \lambda_A + r \cdot (\lambda_B - \lambda_A)$
        if $f(\vec{a} + \lambda_1 \vec{u}) > f(\vec{a} + \lambda_2 \vec{u})$ then
                $\lambda_A \leftarrow \lambda_1$
        else
                $\lambda_B \leftarrow \lambda_2$
        end if
        $\lambda \leftarrow \frac{\lambda_A + \lambda_B}{2}$
```

$s \leftarrow |\lambda_B - \lambda_A|$

`while` $s > \varepsilon_\lambda$ `and` $l < L_{max}$

4 Example results

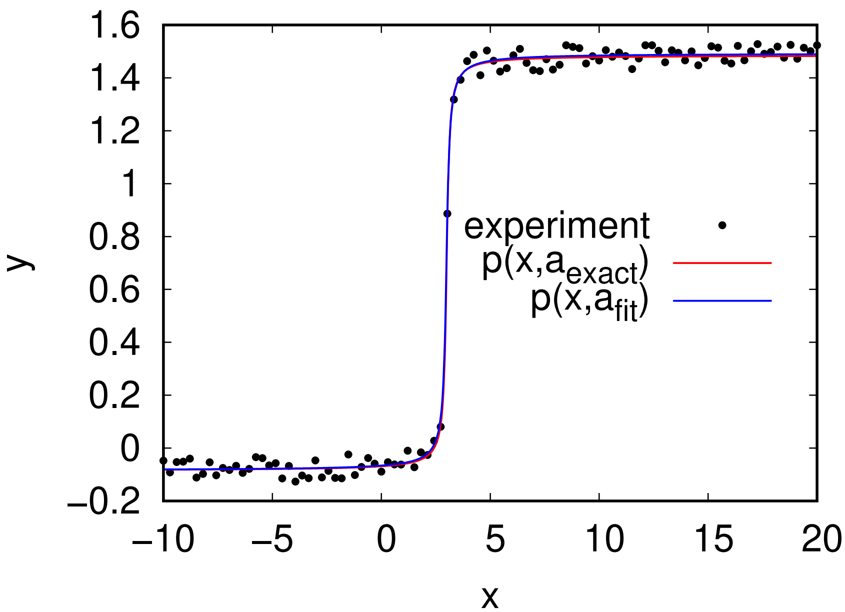

Figure 1: Comparision of "experimental" scattered data (dots) with an exact solution (red) and a fitting function (blue) obtained from minimization routine.