1 Nonlinear Least Squares: Fitting a Lorentz Profile to Data.

Many physical processes produce a Lorentz profile. We are thus often faced with data which we feel should be well modeled by a Lorentzian, and we want to determine the Lorentzian which best fits our data. The Lorentz profile $L(x)$ can be written

$$L(x) \; = \frac{h}{1 \; + \; \left[\frac{2}{w}\,(x-x_0)\right]^2} \tag{1}$$

where $L(x)$ is specified by three parameters:

(1) the central frequency x_0

(2) the height, h, which is the value of $L(x_0)$.

(3) the width, w, which is the full width at half maximim (FWHM) of the profile. (Note that for x such that $2(x-x_0)=w$, $L(x)=h/2$.)

Define $\alpha = 2/w$. Then the profile $L(x)$ is

$$L(x) \; = \frac{h}{1 \; + \; \alpha^2\,(x-x_0)^2} \tag{2}$$

Suppose we have N data points $d_1, d_2, ...d_N$ at frequencies $x_1, x_2, ...x_N$. Then we want to minimize the sum

$$S(x_0, h, w) \; = \; \frac{1}{2} \sum_{i=1}^{i=N} \; (L(x_i) - d_i)^2 \tag{3}$$

where we scale by 1/2 to simplify the expressions. If we define $f_i(x) \; = \; (L(x_i) - d_i)$ we can define the column vector $F(x)$ as

$$F(x) \; = \begin{pmatrix} L(x_1) - d_1 \\ L(x_2) - d_2 \\ \vdots \\ L(x_N) - d_N \end{pmatrix} \; = \; \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_N(x) \end{pmatrix}$$

So then

$$S(x_0, h, w) \; = \; F(x)^T \; F(x) \; , \tag{4}$$

where $F(x)^T$ is the (row vector) transpose of $F(x)$. Now we are interested in the gradient of S, which by the chain rule is

$$\nabla S(x) \; = \; \nabla F(x) \; F(x) \tag{5}$$

Further, the Laplacian $\nabla^2 S(x)$ results from differentiating with respect to x_i:

$$\nabla^2 S(x) \; = \; \nabla F(x) \; \nabla F(x)^T \; + \; \sum_{i=1}^{i=N} \; f_i(x) \; \nabla^2 f_i(x) \tag{6}$$

Equations (5) and (6) are the gradient and Hessian of S.

How do we proceed to find the minimum of $S(x)$?. This will be where the derivative of S is zero. Suppose we have a function $g(z)$. Let us linearize it about some point z_0:

$$g(z) \;=\; g(z_0) \;+\; g'(z_0)(z - z_0) \tag{7}$$

Then to reach the point where $g(z) = 0$, we take a step to z_1:

$$0 \;=\; g(z_0) \;+\; g'(z_0)(z_1 - z_0) \quad\rightarrow\quad \delta z \;=\; (z_1 - z_0) \;=\; -\,\frac{g(z_0)}{g'(z_0)} \tag{8}$$

This is the formula for Newton's method for one variable. We may generalize it to our three (on any number) variables as follows. The function we are solving for is $g \rightarrow \nabla S(x)$:

$$v \;=\; -\,\frac{\nabla S(x)}{\nabla^2 S(x)}\ , \tag{9}$$

where $v \;=\; (\delta x_0, \delta h, \delta w)$ is the correction to the current parameters $x_0,\ h,$ and w. Since the second derivatives are often expensive to compute, it is usual to neglect the 2nd term of equation(6) since this term will vanish close to the solution. We thus make the approximation

$$\nabla^2 S(x) \;\cong\; \nabla F(x)\ \nabla F(x)^T \tag{10}$$

Then our formula for the updated parameters becomes the matrix expression

$$\left(\nabla F(x)\ \nabla F(x)^T\right)\ v \;=\; -\,\nabla F(x)\ F(x) \tag{11}$$

which is a system of linear equations for the vector of corrections v. This is known as the Gauss-Newton method. To evaluate this expression for the Lorenntzian profile, we will need the partial derivatives of L with respect to x_0, h and w:

$$\frac{\partial L}{\partial x_0} \;=\; \frac{2\alpha^2}{h}\ (x - x_0)\ L^2 \tag{12}$$

$$\frac{\partial L}{\partial h} \;=\; \frac{L}{h} \tag{13}$$

$$\frac{\partial L}{\partial \alpha} \;=\; \frac{-2\alpha}{h}\ (x - x_0)^2\ L^2 \tag{14}$$

Since $\frac{\partial L}{\partial w} \;=\; \frac{\partial L}{\partial \alpha}\frac{\partial \alpha}{\partial w}$ and $\frac{\partial \alpha}{\partial w} \;=\; -\,\frac{\alpha^2}{2}$ equation (14) is just

$$\frac{\partial L}{\partial w} \;=\; \frac{\alpha^3}{h}\ (x - x_0)^2\ L^2 \tag{15}$$

Since the data d_i are constants, the gradient $\nabla F(x)$ is thus

$$\nabla F(x) \;=\; \begin{pmatrix} \frac{2\alpha^2}{h}\;(x-x_0)\;L^2 \\ \frac{1}{h}L \\ \frac{-2\alpha}{h}\;(x-x_0)^2\;L^2 \end{pmatrix}$$

We thus evaluate the matrix products $A \;=\; \nabla F(x)\,\nabla F(x)^T$ and $B \;=\; -\nabla F(x)\,F(x)^T$, and then we need to solve $A\;v \;=\; B$ for v.

This procedure is easy to impliment in J, where the solution is just coded by the expression $v \;=\; B$ %. A . The resultant v provides an update to our initial or current x_0, h, w triplet. Finally, we can use J's power verb (:^_) to iterate to convergence.

See the J code "Solve.ijs" on this webpage.

2 Reference

https://math.gmu.edu/ igriva/book/Appendix