Subject: Re: solving alghorithm for gaus curves
Posted by Peter Mason on Fri, 21 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

On Tue, 18 Feb 1997, R. Bauer wrote:
> The next program we want to write will be a flexible fit alghorithm
> tool.
> In the first step we want to start with gaussian curves.
> Is in this group any experience which I can use?

I have an IDL/C application which fits gaussian troughs (absorptions) to
SWIR (short wavelength infrared) mineral reflectance spectra.
Unfortunately the code is not public domain and I can't give it away.
But I'll mention some problems I ran into:

Gaussian fitting is a non-linear optimisation problem and can't be done "in one go" - e.g., you can't use a (direct) linear least-squares algorithm for the job. These non-linear methods are much slower than linear ones because they have to iterate towards a solution. What's worse, you usually need to start them off reasonably close to the optimal solution, otherwise they can easily converge on a non-optimal solution.

My spectra each have hundreds of channels (typically around 600), and I usually want to fit 20 to 30 Gaussians to each spectrum. Solving this is very CPU-intensive, so I chose to implement the non-linear optimiser in C rather than IDL. (An IDL-only version would be far too slow for my particular problem.)
I used the LM (Levenberg-Marquardt) optimising algorithm.
** I gather that this algorithm will be implemented in IDL 5, and that **
** the actual guts of the routine will be "native code" - i.e., fast! **
An issue with the LM algorithm is that you have to decide when it should stop, and there are no hard-and fast rules for determining this, as I understand it.
I chose to implement four "persistence levels" (user selects one) based on the overall fit error and the absolute and relative improvement in an iteration.
(I set the convergence parameters for each level by trial-and-error.)
The LM algorithm can fail due to a singular fitting matrix (aka curvature or covariance matrix?). e.g., the algorithm in Numerical Recipes uses Gauss-Jordan elimination to solve for this matrix, and it will fail if the matrix is singular. (If you implement it, use a singular-value-decomposition solver instead!) You should also check out last year's posts to this news group - Amara Grapps gave some advice on how to improve the LM algorithm.

The main problem I had was finding a good starting solution. Without one, you're *absolutely* wasting your time if you want to fit several gaussians.
I used a "hull featuregram" feature-extraction algorithm as a starting point.
This is probably quite specific to my problem. (I fit only negative gaussians - for absorption features - and fit the "spectral background" separately. The Hull algorithm extracts the background, and the hull featuregrams are all the

same sign.) This algorithm could probably be modified to work without background removal, and to allow mixed-sign features. If you're interested, there's an outline of the algorithm in:
A.A. Green and M.D. Craig, "Analysis of Aircraft Spectrometer Data with Logarithmic Residuals" in "Proceedings of the Airborne Imaging Spectometer Data Analysis Workshop", JPL publication 85-41, June 1995.

Peter Mason