Subject: solving alghorithm for gaus curves
Posted by R. Bauer on Tue, 18 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Hi

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?

--
R.Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@kfa-juelich.de

Subject: Re: solving alghorithm for gaus curves
Posted by rivers on Wed, 19 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

In article <33099F87.5E2@nv.et-inf.uni-siegen.de>, Achim Hein <hein@nv.et-inf.uni-siegen.de>
writes:
> R. Bauer wrote:
>>
>> Hi
>>
>> 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?
>>
>> --

We use curvefit to fit Gaussian peaks in spectra with good success.

Mark Rivers (773) 702-2279 (office)
CARS (773) 702-9951 (secretary)
Univ. of Chicago (773) 702-5454 (FAX)
5640 S. Ellis Ave. (708) 922-0499 (home)
Chicago, IL 60637 rivers@cars.uchicago.edu (e-mail)

or:
Argonne National Laboratory (630) 252-0422 (office)
Building 434A (630) 252-0405 (lab)
9700 South Cass Avenue (630) 252-1713 (beamline)
Argonne, IL 60439 (630) 252-0443 (FAX)

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

In article <Pine.SUN.3.91.970221120406.3491B-100000@demsyd.syd.dem.csiro.au>, Peter Mason <peterm@demsyd.syd.dem.csiro.au> writes:
> 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?
>
>
> 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 routinely work on similar scale problems with energy-dispersive x-ray fluorescence data. There are 2048 channels of data and 10-30 peaks to fit. I used to use CALL_EXTERNAL to an IMSL fitting routine, but have switched to using CURVEFIT in IDL. That way the application is portable and an IMSL license is not required. The performance hit was only about a factor of 2. Fitting a spectrum on a low-end DEC Alpha takes about 5-10 seconds. We also fit the background separately.

In general when fitting multiple Gaussians there are 3 parameters to be fit for each peak: centroid, width and amplitude. In certain applications it may make sense to constrain one or more of these. For example, when fitting our XRF data, the position of each peak is typically not optimized, since the fluorescence energies are known and constant. Rather, only 2 energy calibration coefficients (which control the relation of channel # to energy) are fitted. Similarly, I know the instrument response function of my detector is sigma=A + B*SQRT(energy). Thus sigma of each peak is typically not fitted

independently, but rather only the coefficients A and B are optimized.

Making use of the physics of the experiment not only speeds things up, but makes for results which are more physically meaningful.

Mark Rivers (773) 702-2279 (office)
CARS (773) 702-9951 (secretary)
Univ. of Chicago (773) 702-5454 (FAX)
5640 S. Ellis Ave. (708) 922-0499 (home)
Chicago, IL 60637 rivers@cars.uchicago.edu (e-mail)

or:
Argonne National Laboratory (630) 252-0422 (office)
Building 434A (630) 252-0405 (lab)
9700 South Cass Avenue (630) 252-1713 (beamline)
Argonne, IL 60439 (630) 252-0443 (FAX)

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

Subject: Re: solving alghorithm for gaus curves
Posted by peter on Sun, 23 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Mark Rivers (rivers@cars3.uchicago.edu) wrote:
: I routinely work on similar scale problems with energy-dispersive x-ray
: fluorescence data. There are 2048 channels of data and 10-30 peaks to fit. I
: used to use CALL_EXTERNAL to an IMSL fitting routine, but have switched to
: using CURVEFIT in IDL. That way the application is portable and an IMSL
: license is not required. The performance hit was only about a factor of 2.
: Fitting a spectrum on a low-end DEC Alpha takes about 5-10 seconds.
: We also fit the background separately.

: In general when fitting multiple Gaussians there are 3 parameters to be fit for
: each peak: centroid, width and amplitude. In certain applications it may make
: sense to constrain one or more of these. For example, when fitting our XRF
: data, the position of each peak is typically not optimized, since the
: fluorescence energies are known and constant. Rather, only 2 energy calibration

: coefficients (which control the relation of channel # to energy) are fitted.
: Similarly, I know the instrument response function of my detector is
: sigma=A + B*SQRT(energy). Thus sigma of each peak is typically not fitted
: independently, but rather only the coefficients A and B are optimized.

: Making use of the physics of the experiment not only speeds things up, but
: makes for results which are more physically meaningful.

To follow up Mark's comment: once you are down to amplitude only, the problem becomes linear again, and can be solved without iteration. It often pays, if you have a pretty good idea of the non-linear parameters, but no idea of the linear ones (e.g. here you know the widths, but not the amplitudes) to fix the non-linears, perform a linear fit to get the amplitudes, then start the non-linear optimizer at a good starting point.

Peter
