Subject: Re: Help Wanted: IDL Math Expert
Posted by John Votaw on Tue, 19 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Hi Folks,
>
> I had a friend ask a question about IDL that I didn't know how
> to answer. (Uh, math is not my strength, you understand. :-))
>
> Suppose I have a set of raw data that is described
> theoretically by two parametric equations. And suppose
> I need to fit these two parametric equations to the data
> since there is no mathematical way to convert the x(t)
> and y(t) equations into a y(x) form. Is there a
> curve fitting routine in IDL that can handle parametric
> equations? Or, failing that, has anyone handled something
> like this in IDL and would be willing to give us a little
> help?
>> Many thanks,

The real question here is what do you mean by a good fit? You must
write an equation in terms of the parameters in your parametric
equations that is minimal when the desired fit is achieved.

For example, suppose your data is (x,y) pairs and you would like to fit
it to a generalized ellipse. Do you want to minimize the squared
distance along a vector from the center of the ellipse between the data
and the fit? Perhaps you want to minimize the distance along a line
normal to the fit curve.

The difficult part is writting this 'figure of merit' equation. Once
you have it, you can use any of the function minimization routines. I
suggest you start by looking at Powell.

Good Luck,

John R. Votaw
votaw@commander.eushc.org

Subject: Re: Help Wanted: IDL Math Expert
Posted by Jack Saba on Tue, 19 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>

Why not treat it as a pair of coupled nonlinear equations and use
one of the standard techinques, e.g., Newton's method?

> Hi Folks,
>
> I had a friend ask a question about IDL that I didn't know how
> to answer. (Uh, math is not my strength, you understand. :-))
>
> Suppose I have a set of raw data that is described
> theoretically by two parametric equations. And suppose
> I need to fit these two parametric equations to the data
> since there is no mathematical way to convert the x(t)
> and y(t) equations into a y(x) form. Is there a
> curve fitting routine in IDL that can handle parametric
> equations? Or, failing that, has anyone handled something
> like this in IDL and would be willing to give us a little
> help?
>
> Many thanks,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting
> Customizable IDL Programming Courses
> Phone: 970-221-0438 E-Mail: davidf@dfanning.com
> Coyote's Guide to IDL Programming: http://www.dfanning.com

--
Jack Saba <jsaba@magus.stx.com>

Subject: Re: Help Wanted: IDL Math Expert
Posted by J.D. Smith on Tue, 19 Aug 1997 07:00:00 GMT

View Forum Message <> Reply to Message

David Fanning wrote:
>
> Hi Folks,
>
> I had a friend ask a question about IDL that I didn't know how
> to answer. (Uh, math is not my strength, you understand. :-))
>
> Suppose I have a set of raw data that is described
> theoretically by two parametric equations. And suppose
> I need to fit these two parametric equations to the data
> since there is no mathematical way to convert the x(t)

> and y(t) equations into a y(x) form. Is there a
> curve fitting routine in IDL that can handle parametric
> equations? Or, failing that, has anyone handled something
> like this in IDL and would be willing to give us a little
> help?
>
> Many thanks,
>
> David
>

Presuming I understand you correctly, here is how I'd handle it: You have some data y and x, each of which depend on a parameter t through the equations y=y(t); x=x(t). Neither are analytically invertible..
i.e. you cannot compute:

```
   -1
t=x (t)[x] ; y=y(t)
```

where the inverse of x(t) is applied to the data values x. The solution is to numerically compute the inverse of x(t) for each data value x. There's no efficient IDL builtin for doing this, though NEWTON could be used. I'd use zbrent() based on Num. Recipes routine of the same name. It's in the nasa package of IDL routines, available many places. This would be called in the curvefit fitting function. E.g.

```
fit=curvefit(xdata,ydata,weights, A, /NODERIVATIVE,FUNCTION_NAME='func')
```

where A=[ax,bx,cx,...,ay,by,cy,...] ... i.e. the parameters from both your functions concatenated together. And now for func:

```
pro func,X,A,F
 common funcblock, Asav, Xsav
 Asav=A & Xsav=X
 tmin=0. & tmax=200. ; substitute min and max possible t's
      ; x(tmin) and x(tmax) must be opposite sign
      ; these values might depend on X, or could
       ; be fixed (valid for each X data value)
 t=zbrent(tmin,tmax,FUNC_NAME='brentfunc')
 F=.... ;compute F=y(t) with t and ay,by,cy,....
 return
end

function brentfunc,t
 common funcblock, Asav, Xsav
 x= ... ;compute x(t) using ax,bx,cx,... from Asav
 return, x-Xsav ;finds t such that x(t)=Xsav (data value)
```

end

This should do it. It's a bit inefficient. If x and t increase
together for all relevant t, you could make it more efficient by
inverting the xdata to t in a separate step, using the previous value of
t found as the new tmin (a starting point) in the call to zbrent for the
next X (sorted of course). This should save a few iterations in zbrent
(but it's usually really fast), and overcome curvefit inefficiencies.
Good luck.

JD

Subject: Re: Help Wanted: IDL Math Expert
Posted by Brad Gom on Tue, 19 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Dr. Martin Ryba wrote:
>
> It sounds like the "raw data" has nonzero uncertainties in
> both the X and Y direction. Normal least-squares techniques assume a
> negligible error in the "independent variable" and are forming a

I'm assuming that the data has no error.

> If one can generate the T variable for
> each X,Y pair (which is kinda implied by your question), then I would
> concatenate the X and Y data together, each with its own copy of T, and
> use CURVEFIT to fit the combined data set with the concatenated set of
> free parameters.

I thought of doing this but I don't think I can generate the T variable
from the raw data.

To be more precise, my data is described by x(t)=sqrt(F/G) and
y(t)=sqrt(FG)
where F=a*((t^2)/b - t) and G=c*exp(sqrt(d/t))
I want the values of a,b,c, and d

I thought the following might work:
1. estimate parameters
2. calculate both x(t) and y(t) for range of t
3. compute difference between [x(t) vs y(t)] and the raw data
4. adjust parameters and return to 2. until fit

I could easily write this code except for step number 4. Would it be

possible to modify the curvefit
procedure to perform this step? I'm hoping I don't have to take a course
on nonlinear curve fitting to
solve this problem.

Thanx for the help

Brad Gom

Subject: Re: Help Wanted: IDL Math Expert
Posted by Marty Ryba on Tue, 19 Aug 1997 07:00:00 GMT

View Forum Message <> Reply to Message

David Fanning wrote:

> Suppose I have a set of raw data that is described
> theoretically by two parametric equations. And suppose
> I need to fit these two parametric equations to the data
> since there is no mathematical way to convert the x(t)
> and y(t) equations into a y(x) form. Is there a
> curve fitting routine in IDL that can handle parametric
> equations? Or, failing that, has anyone handled something
> like this in IDL and would be willing to give us a little
> help?

Hmmm. The short answer to this question is "It Depends." (I know, lotta help). It sounds like the "raw data" has nonzero uncertainties in both the X and Y direction. Normal least-squares techniques assume a negligible error in the "independent variable" and are forming a maximum-likelihood solution given weights and uncertainties in the Y (dependent variable) direction. If one can generate the T variable for each X,Y pair (which is kinda implied by your question), then I would concatenate the X and Y data together, each with its own copy of T, and use CURVEFIT to fit the combined data set with the concatenated set of free parameters. The user-supplied function would keep track of which points are X and Y (first half and second half, or interleaved), and some of the partial derivatives would be simply zero. The weights for each X,Y data point would be proportial to the inverse of the error squared or whatever else you'd like to try.

Good luck; nonlinear least-squares fitting is part science and part art.
If you need a more full-featured fitting routine, try SUPERFIT in I
believe the JHU/APL astrophysics library.

Dr. Marty Ryba
MIT Lincoln Laboratory
ryba@ll.mit.edu

Subject: Re: Help Wanted: IDL Math Expert
Posted by David Ritscher on Wed, 20 Aug 1997 07:00:00 GMT

View Forum Message <> Reply to Message

Brad Gom wrote:
> I'm assuming that the data has no error.
...

> To be more precise, my data is described by x(t)=sqrt(F/G) and
> y(t)=sqrt(FG)
> where F=a*((t^2)/b - t) and G=c*exp(sqrt(d/t))
> I want the values of a,b,c, and d

> I thought the following might work:
>   1. estimate parameters
>   2. calculate both x(t) and y(t) for range of t
>   3. compute difference between [x(t) vs y(t)] and the raw data
>   4. adjust parameters and return to 2. until fit

> I could easily write this code except for step number 4. Would it be
> possible to modify the curvefit
> procedure to perform this step? I'm hoping I don't have to take a course
> on nonlinear curve fitting to
> solve this problem.

Perhaps I'm missing something here, but it seems like you'd just want
to detangle the two functions, which is trivial, in the noiseless
case, with the following calculations:

F = x(t) * y(t)
G = y(t) / x(t)

Then you do the two straight-forward fittings.

I hope this helps,

Regards,

David Ritscher

--
David Ritscher
Zentralinstitut fuer Biomedizinische Technik
Albert-Einstein-Allee 47   Tel: +49 (731) 502 5313
Universitaet Ulm   Fax: +49 (731) 502 5315
D-89069 ULM   david.ritscher@zibmt.uni-ulm.de
Germany