Subject: optimization using IDL
Posted by Karl Young on Mon, 16 Sep 1996 07:00:00 GMT
View Forum Message <> Reply to Message

I know this has been discussed before and I apologize for any
redunancy. I am finding IDL's curvefit inadequate for the types
of optimization I need to do. For one thing I need to do
constrained optimization. I hacked curvefit to do a kludgy
version of constrained optimization and a coleague put
together an even better version but I'm at the point at which
I really need an "industrial strength" constrained optimizer.
(I went through the section in Bevington's book from which
IDL's curvefit was culled and there didn't seem to be any really
obvious ways to extend that algorithm to do the kind of
constrained optimization that you find in modern textbooks on
optimization, i.e. I generally couldn't find any way of
extending standard Marquardt-Levinson type optimization, but
maybe I'm reading the wrong books)

So my question is; has anybody been in this situation with
IDL and if so how did they deal with it ? E.g. has anybody
written more sophisticated optimization code in IDL, do
people generally call external Fortran or C routines from
commercial or shareware packages, ... ? Any comments or
suggestions greatly apreciated,

-- Karl Young   karl@gojira.berkeley.edu

Subject: Re: optimization using IDL
Posted by Amara Graps on Tue, 17 Sep 1996 07:00:00 GMT
View Forum Message <> Reply to Message

Karl Young wrote:
>
> I know this has been discussed before and I apologize for any
> redunancy. I am finding IDL's curvefit inadequate for the types
> of optimization I need to do. For one thing I need to do
> constrained optimization. I hacked curvefit to do a kludgy
> version of constrained optimization and a coleague put
> together an even better version but I'm at the point at which
> I really need an "industrial strength" constrained optimizer.
> (I went through the section in Bevington's book from which
> IDL's curvefit was culled and there didn't seem to be any really
> obvious ways to extend that algorithm to do the kind of
> constrained optimization that you find in modern textbooks on
> optimization, i.e. I generally couldn't find any way of
> extending standard Marquardt-Levinson type optimization, but

> maybe I'm reading the wrong books)

Hi Karl,

Do you want to stick with the Marquardt algorithm or try another algorithm? I'm aware of more sophisticated optimization techniques (but have never needed them, and would only be able to give references).

If you want to stick with the Marquardt algorithm, another thing to try is to work with the lambda parameter. The Bevington/Numerical Recipes/IDL routine uses a lambda that is the same value for all of the free parameters. Make lambda an _array_ instead, with a slightly different value for each of the free parameters.

If you are navigating a tricky Chi-square "terrain" where particular parameters give really flat chi-squares and therefore, the iterations go on forever, this technique helps one get out of that problem. One would have to have a pretty good understanding of their function that they are trying to fit, however.

For example, one could do something like the following:

```
    ;percentage of lambda to increase/decrease
    del_flambda = [30.d0,30.d0,30.d0,30.d0,25.d0,25.d0]
```

And so inside of the CURVFIT routine, the flambda looks like:

```
    flambda = flambda/del_flambda
```

You can also impose your your own min and max inside the CURVFIT routine.

For example, let's say you have 6 free parameters v(0..5). You could initially set up the minimums/maximums for them:

```
   ;Min/max values allowed for v()
   minmax(0,0)=[v(0)/3,v(1)-3*cf,v(2)/3,.0015,v(4),0]  ;min
   minmax(0,1)=[v(0)*3,v(1)+3*cf,v(2)*3,.0027,v(4),0]  ;max

   mmn=2   ;Least #iterations to impose temporary minmax values
   mmx=100 ;Max #iterations to impose temporary minmax values
```

(This is just an example of min/max's I've used for a curvefitting problem a few years ago.)

Then use these inside of CURVFIT in the iterations loop, checking to see if the value of the parameters have exceeded the min/max's.

The Marquardt curvefitting is also extremely sensitive to initial guesses. That's another place where one's knowledge of their curvefitting function is necessary. If one doesn't have good initial guesses, they're in trouble. I've recently thought that a genetic algorithm method might work well for coming up with good initial guesses, but have never implemented anything like that.

Are your derivatives accurate? That is a really easy place for one to make a mistake, and will make the Marquardt algorithm completely useless.

So I don't know if the above techniques are is the "industrial strength" Marquardt curvefitting, that you were looking for, (and I'm not even sure if these tricks make sense mathematically) but these tricks worked pretty well for me fitting some complicated functions that consisted of integrals, numerical derivatives, and with up to 7 free parameters.

Amara

--

*** *****
Amara Graps amara@quake.stanford.edu
Solar Oscillation Investigations Stanford University
http://quake.stanford.edu/~amara/amara.html
*** *****
"Never fight an inanimate object." - P. J. O'Rourke

Subject: Re: optimization using IDL
Posted by rivers on Tue, 17 Sep 1996 07:00:00 GMT

View Forum Message <> Reply to Message

In article <323DE338.41C67EA6@gojira.berkeley.edu>, Karl Young <karl@gojira.berkeley.edu> writes:
> I know this has been discussed before and I apologize for any
> redunancy. I am finding IDL's curvefit inadequate for the types
> of optimization I need to do. For one thing I need to do
> constrained optimization.
...
> So my question is; has anybody been in this situation with
> IDL and if so how did they deal with it ? E.g. has anybody

> written more sophisticated optimization code in IDL, do
> people generally call external Fortran or C routines from
> commercial or shareware packages, ... ? Any comments or
> suggestions greatly apreciated,

I have worked with RSI to fix some of the limitations in curvefit (e.g. it now longer needs analytical derivatives). For more sophisticated optimization I have used CALL_EXTERNAL and the IMSL or NAG libraries.

Mark Rivers	(312) 702-2279 (office)
CARS	(312) 702-9951 (secretary)
Univ. of Chicago	(312) 702-5454 (FAX)
5640 S. Ellis Ave.	(708) 922-0499 (home)
Chicago, IL 60637	rivers@cars3.uchicago.edu (Internet)

Subject: Re: optimization using IDL
Posted by PREUSSER on Fri, 20 Sep 1996 07:00:00 GMT

View Forum Message <> Reply to Message

In article <323DE338.41C67EA6@gojira.berkeley.edu>, Karl Young <karl@gojira.berkeley.edu> wrote:

> I know this has been discussed before and I apologize for any
> redunancy. I am finding IDL's curvefit inadequate for the types
> of optimization I need to do. For one thing I need to do
> constrained optimization. I hacked curvefit to do a kludgy
> version of constrained optimization and a coleague put
> together an even better version but I'm at the point at which
> I really need an "industrial strength" constrained optimizer.
> (I went through the section in Bevington's book from which
> IDL's curvefit was culled and there didn't seem to be any really
> obvious ways to extend that algorithm to do the kind of
> constrained optimization that you find in modern textbooks on
> optimization, i.e. I generally couldn't find any way of
> extending standard Marquardt-Levinson type optimization, but
> maybe I'm reading the wrong books)
>
> So my question is; has anybody been in this situation with
> IDL and if so how did they deal with it ? E.g. has anybody
> written more sophisticated optimization code in IDL, do
> people generally call external Fortran or C routines from
> commercial or shareware packages, ... ? Any comments or
> suggestions greatly apreciated,
>
> -- Karl Young   karl@gojira.berkeley.edu

Karl,
I have to make two comments on your request:

1) I have developped a curve fitting program running with PV-Wave
Advantage that allows generating starting values close to
the desired solution, so that often you do not need to apply
special constraints. At the moment, it is limited to fit with
a sequence of Gaussians and a quadratic polynomial. But you
are able to keep any parameter constant, that is, force it to
a certain value. The called solver is that of PV-Wave Advantage,
the Levenberg-Marquardt from Minpack. Anybody interested can
download the latest version from the following address:
http://www.fhi-berlin.mpg.de/~grz/pub/preusser.html

2) If you still need to use a solver with constraints of "industrial
strength", I recommend you to contact
WTI@aol.com or http://members.aol.com/WTI
They are a company specializing in constrained optimization, and
offer industrial solution at resonable prices. It is a group
split from IMSL.

Hope that will help you and others interested in this topic.

--
Dr. Albrecht Preusser | Basic Engineering:
Gemeinsames Rechenzentrum | ---------------------
Fritz-Haber-Institut der MPG | Nothing for nothing
Faradayweg 4-6 | Nothing for everything
D-14195 Berlin (Dahlem) | Nothing for ever
Phone: +49-30-8413-3220 | ---------------------
http://www.fhi-berlin.mpg.de/~grz/pub/preusser.html