

---

Subject: Re: General fitting framework

Posted by [Rob.Dimeo](#) on Wed, 25 Feb 2009 21:09:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

This should be pretty straightforward to write (having gone through a similar exercise some years ago...no, I'm not volunteering). The key point is making the interface to the fitting function (e.g. MPCURVEFIT, GA\_FIT, PSO\_FIT, etc.) sufficiently general (i.e. identical) so that the user can specify the fitting method through a GUI interface, for instance, and then using CALL\_FUNCTION or CALL\_PROCEDURE for the fitting. I tried to do just this with a particle swarm optimization algorithm and the genetic algorithm optimizer. I forced both of these functions to have a similar interface and keywords as Craig Markwardt's MPCURVEFIT. Then they could be called in the same way as Craig's algorithm.

You don't really need to use objects to do this but I've found them to be pretty useful in making a general-purpose fitting application, particularly with handling the fitting components and data where the data is multi-dimensional.

Rob

On Feb 25, 1:03 pm, Giorgio <giorgiol...@gmail.com> wrote:

- > Hi,
- > For my research work, the data analysis is often reduced to fit data
- > with a model. For that I often use MPfit which is a basically
- > Levenberg-Marquardt algorithm. In general this performs well however I
- > found that it depends a lot on the initial guess of the parameters and
- > it often gets stuck in a local minima. There are other open source
- > optimization algorithms written in IDL like genetic algorithms (Rob
- > Dimeo's code) or simulated annealing (Kling's book). I am thinking in
- > a way to implement a general fitting program that allow you to choose
- > which algorithm to use. The best way I thought it will be by using
- > object oriented programming. The different class objects can be:
- > - A class to hold the data, this can be either 2D or 1D data.
- > - A class to perform the function evaluations. The way that mpfit
- > treats the function is quite general. This would have the function
- > string and the number of parameters supported by the function. One
- > method will be the evaluation of the function with the values
- > positions where to do it.
- > - A function to evaluate the difference between the data and the
- > model function. This is usually the chi-squared function but it can
- > also be another one like the sum of the absolute difference. I did not
- > know yet which if the best approach is a function or an object.
- > - A class for the parameters to handle the parinfo structure define
- > in the mpfit program.
- > - And finally, a class to define a general framework for the

> optimization algorithms. Each precise algorithm will be a child of  
> this class and will contain the algorithms parameters in it.  
> This is the general framework I thought about. By looking at this I  
> realized that it will take a lot of time for me alone to do it since I  
> will do it in my spare time. So my questions will be: Are there any  
> suggestions or ideas of which will be a better implementation of this?  
> And is there someone who will like to contribute or help me in doing  
> this?  
>  
> Thanks,  
>  
> Giorgio

---