
Subject: FIT across images

Posted by [bas.basetta](#) on Fri, 27 Mar 2009 15:28:22 GMT

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

Hi to everybody.

I used this group so many times as a place where to find solutions to IDL problems, and I think it is maybe the most useful and accurate resource to do that, you are great. I never posted, because most of the questions have already been asked and answered. I'm doing this now because I have one that can be of general interest, and it is about fitting.

Chloe' is asking:

http://groups.google.co.uk/group/comp.lang.idl-pvwave/browse_thread/thread/a4b97b57a2f194f7/

I don't know if mine is the same, but I'm trying to explain it the best I can.

I'm using Craig Markwardt's legendary MPCURVEFIT, and I have a set of n images, each being a $\text{size} \times \text{size}$ matrix. I want to fit an exponential (or in general a function) through those images, i.e. each pixel is fitted across n images to produce an exponential pixel by pixel map. Now - disgusted by myself :) not finding a proper solution - I'm repeating the fit process $\text{size} \times \text{size}$ times:

```
For i=0 to size, For j=0 to size
  fit=mpcurvefit(x,S,W,A,sigma,function_name='expf',chisq=chis q,/
  noderivative)
```

x is the variable (a parameter which is the same for the whole image) changing across images and S is the luminance of each pixel. $x[n]$, $S[n]$ and $W[n]$ are vectors and have dimension n and A is a vector with dimension [3].

'expf' is defined

```
bx = EXP(-x*A[1])
F = A[0] * bx + A[2]
```

and then I store A for each pixel

```
map=[A[1],i,j]
```

It works fine, but it is very time consuming, and I suspect it is not very efficient in a array-oriented language. I would really like the whole image to be processed at each iteration.

This should be done using:

```
fit=mpcurvefit(x,S,W,A,sigma,function_name='expmap',chisq=chisq,  
noderivative)
```

where $x[n]$ is the same, but now $S=[n,\text{size},\text{size}]$, $W=[n,\text{size},\text{size}]$ and $A=[3,\text{size},\text{size}]$

and 'expmap' is defined

```
bx = EXP(-x*A[1,*,*])  
F = A[0,*,*] * bx + A[2,*,*]
```

In principle X and Y should be of any size, provided that a correct function is given to map X to Y . 'expmap' is producing $F=[1,\text{size},\text{size}]$.

The procedure "mpcurvefit" does not complain but also does no iteration, giving me back A unchanged, i.e. initial conditions.

First I thought it could be a problem of coherent array's dimension, and I tried to replicate $x[n] \rightarrow x[n,\text{size},\text{size}]$, with similar results.

I played with dimensions, and if something is set wrong, it complains.

But I still think that I'm not passing the parameters in the correct way, because the resulting variable is $\text{fit}[n]=S[^*,0,0]$, i.e. the first point on n images.

Or maybe I'm missing some evident principle because the routine is not meant to work like that, but I'm realistically thinking that I'm not able to use it properly. Obviously it would be great also a solution where I can use $S[n,\text{size}*\text{size}]$, i.e. a very long array built up with the image's rows, if the problem is the 3rd dimension.

I'm also trying to figure out if the general principle is not working: if it is trying to minimize a parameter (chisq) how could it find one representative of the whole image at each iteration, and meet a convergence criteria? Maybe that's the point.

I need some help in case I'm using Craig's routine incorrectly, or some suggestion, because I'm struggling, and maybe I'm missing a BIG thing, evident for experts like you all are.

Many thanks in advance,
Bas
