Subject: Re: fitting many linear eqs simultaneously with outliers
Posted by Craig Markwardt on Sat, 15 Oct 2011 01:46:56 GMT
View Forum Message <> Reply to Message

On Oct 13, 1:01 am, Jeremy Bailin <astroco...@gmail.com> wrote:
> So I have a large number of very simple linear equations, which all look
> like:
>
> a_i + b_i x_ij = a_k + b_k x_kj
>
> for i,k=1..N (all unique combinations of i and k), j=1..M. The data
> values x_ij and x_kj are measurements of the brightness of object j in
> images i and k respectively, where the images have an unknown zero point
> and scalings that I am trying to determine.
>
> (aside to astronomers: this may sound suspiciously like re-implementing
> mscimatch)
>
> Not all objects appear in all images, so there are different numbers of
> equations relating each pair of images. In principle, any number of
> linear solvers should work... BUT: it needs to be extremely robust to
> outliers. I know for a fact that there are many many many outliers, and
> there are some pairs of images where it looks like pure scatter. So I
> need some sort of solver that will do sigma clipping. Essentially, I
> want a sigma-clipping linear least squares solver that can solve more
> than one line at once.
>
> Does anyone know of such a beast already existing? Or something that's
> vaguely similar enough that I can use it as a basis?

I'm pretty sure you can use MPFIT to solve this set of equations (MPFIT, not MPFITFUN). You need to rephrase the equations trivially as LEFT - RIGHT = 0, and then MPFIT will happily solve all of these equations in a least squares sense. Your user function computes LEFT-RIGHT for each equation. Presumably you would want to scale by the uncertainties in each equation as well so that (LEFT-RIGHT)/SCALE has an expected variance of 1.

As for outliers, I have used a TANH() filter in the past. In other words, solve this slightly modified problem,

NSIGMA*TANH((LEFT-RIGHT)/(NSIGMA*SCALE)) = 0

TANH() has the property of being linear near the origin and "truncating" smoothly values much greater than 1. The NSIGMA part is a N-sigma filter, i.e. if NSIGMA=3 then 1- and 2-sigma variations should pass through relatively unscathed, but 3 to 100 sigma outliers would be stomped down to about 3 sigma.

If it were my preference, I would re-express the problem so that your model function predicts the measured intensity of each source in each plate. For example, using MPFITFUN and this model function,

X_IJ_MODEL = C_I + D_I * F_J

where C_I and D_I are slightly different formulations of your zero-point and scale for the Ith image, and F_J is the "true" relative flux of the Jth source. The F_J are nuisance parameters (and you need to set F_0 = 1 to prevent degeneracy). I see this as better because you probably have measurement uncertainties of X_IJ so the problem is likely to be more linear and stable when expressed this way.

Good luck,
Craig
