Re: Fitting Functions

Source: http://sci.tech−archive.net/Archive/sci.physics/2004−06/5388.html

---

mmeron_at_cars3.uchicago.edu
Date: 06/16/04

```
Date: Wed, 16 Jun 2004 22:28:31 GMT
```

In article <caqfvpueh1@hood.uits.indiana.edu>, glhansen@steel.ucs.indiana.edu (Gregory L. Hansen) writes:
>In article <iM2Ac.13$25.4207@news.uchicago.edu>,
> <mmeron@cars3.uchicago.edu> wrote:
>>In article <capl9s$he4$1@hood.uits.indiana.edu>,
>>glhansen@steel.ucs.indiana.edu (Gregory L. Hansen) writes:
>>>In article <sQOzc.31$45.13163@news.uchicago.edu>,
>>> <mmeron@cars3.uchicago.edu> wrote:
>>>>In article <cann9o$t4h$1@hood.uits.indiana.edu>,
>>>>glhansen@steel.ucs.indiana.edu (Gregory L. Hansen) writes:
>>>>>
*>>>>>It seems so simple; let the computer fit y0+A*exp(−invtau*x) to a peice of*
>>>>>data and that's that. But depending on the starting conditions I got
>>>>>invtau = 0.246, 0.405, 0.467, and they all looked pretty good. And I fit
*>>>>>to my own function, y0+A*exp(−invtau(x−x0)) with x0 held constant, and got*
>>>>>invtau=0.626. That looks a lot more like the value I expected, so that's
>>>>>what I'll keep. What the hell, it's as good as any other. Lowest
>>>>>chisquare, but the chisquares of the first three only differed by a few
>>>>>percent.
>>>>>
>>>>>When fitting to data like that, how can I have reasonable confidence that
>>>>>my number is the one that corresponds to the real world?
>>>>>
>>>>In general, you cannot. A sufficient amount of beer will improve the
>>>>confidence (ale is better than lager in this respect) and few shots of
>>>>Jack Daniels may work even better, but the effect is transient.
>>>>
>>>>To the point, if your fitting procedure estimates only the fit
>>>>parameters, not their errors as well, then it is not good enough.
>>>
>>>I use Igor Pro, and that will give fits, uncertainties, chi−squares,
>>>covariance matrices, confidence intervals... I don't even know what
>>>confidence intervals are.
>>>
>>Well, you better check. same as with instrumentation, it is
>>worthwhile knowing what the various knobs and buttons are doing.
>>
>>>But to give an example from yesterday, these are four different fits to

>>>the same set of data. The first three were fit to y0+A*exp(−invtau*x),
>>>the fourth to y+A*exp(−invtau*(x−x0)) with x0 given and fixed.
>>>
>>>invtau chi_square
>>>−−−−−−−−−−−−−− −−−−−−−−−−
>>>0.245 +− 5e−13 9.92e−14
>>>0.403 +− 0.154 9.50e−14
>>>0.467 +− 0.163 9.40e−14
>>>0.626 +− 0.194 9.32e−14
>>>
>>>The first point can clearly be discarded because its uncertainty is
>>>unreasonably small; it has to be an artifact of something going wrong in
>>>the fit.
>>
>>Yes, my feelings exactly.
>>
>> The one with the lowest chi−square has the highest uncertainty.
>>>I haven't tried to find an expected standard deviation in chi−squares, but
>>>e.g. the last two differ by 0.85%, which off the cuff I'd have thought is
>>>not a significant difference, while the returned values differ by 30%,
>>>which is huge.
>>
>>Why do you think it is huge? Looking at the table above they appear
>>to be within each other's error bars. What you've here is a situation
>>where the fit quality is nearly independent of the value of this
>>specific parameter, over a broad range. That's why the error bars are
>>so big. The less the chi−square depends on a specific parameter, the
>>poorer is the determination of said paramater.
>>
>>As to why the fit converges to different values, there are two
>>possibilities. One is that you've local minima. To investigate this
>>you would have to generate a sample the function values around such
>>point and look at it (yes, I know, it is 3D, but you can pick 2d cross
>>sections). The other possibility (which seems more likely to me in
>>this case) is that (due to the very slight dependence on nitau) the
>>convergence to the minimum is so slow that the routine simply decides
>>that that's good enough and calls it quits. Most minimization routines
>>use some smallness criterion where when the rate of change (the
>>gradient, to be strict) gets small enough they decide that "it is good
>>enough".
>
>There's ways to adjust all of that. But when there aren't boxes to check
>or fill in, it's easy to forget everything that's there.

That's why I prefer, to the extent possible, to use routines I write
myself. This way I know what's in there. Well, to be exact, I know
it for a while, but coming back to some of them, later, it often takes
a while to figure out what's going on.
>>
>>> And the data is noisy enough that it's not really obvious
>>>from eyeballing it that one is better than the other.

>>>
>>>I'm more inclined to believe the fourth one because of the extra
>>>information I gave it; the heater switched state at x0 = 40 minutes.
>>>The elbow of the exponential can be adjusted by changing either A or
>>>invtau, and the latter fitting function expands to
>>>
>>> y0 + A exp(invtau*x0) exp(−invtau*x)
>>>
>>>and the former's multiplying factor, A'=A*exp(invtau*x0), makes me
>>>nervous because it contains several physically meaningful parameters and
>>>because that exponential turns small variations into big changes.
>>>
>>It is worth than this since it folds few parameters into a single
>>parameter. For any specific value of nitau, a change in A can be
>>nullified by an appropriate change in x0. That's an illposed problem,
>>the answer is no longer unique. I suggest you drop this x0.
>
>No, the x0 is held fixed. I know what it is.

ah, OK. I was under the impression that it is a fit parameter.

Mati Meron | "When you argue with a fool,
meron@cars.uchicago.edu | chances are he is doing just the same"