Subject: Fitting a Gaussian to an "unleveled" histogram: Interpreting the width

Posted by Maryam on Thu, 12 Feb 2015 00:52:08 GMT

View Forum Message <> Reply to Message

Hello.

I am trying to fit an inverted Gaussian function to a histogram that is higher at one end than the other. I simply define a function that is a Gaussian with negative amplitude (plus a constant to shift the Gaussian up along the y-axis) and call that function using mpfitfun to get the fit. It all works out fine and I do get a fit that looks reasonable. However, I am not sure how to interpret the standard deviation returned by mpfitfun. I first thought that IDL uses the lower side to make the fit, but I noticed it is not so. Does anyone know how the Gaussian parameters (in particular, the standard deviation) is calculated by IDL in this case? Please note that I fit the Gaussian to the midpoint of each bin whose coordinates are (xhist, yhist).

Thanks,
Maryam
P.S. See below for the code I wrote. You don't need to have the histogram (i.e. newr) to reproduce what I have (that's why I have commented out that line with plothist). Just plot, xhist, yhist, psym=4 (I printed xhist and yhist arrays below) and follow the steps I have copied below to fit the Gaussian to those points.

```
function gaussianfit, x, p

Amp = p[0]
xc = p[1]
W = p[2]
const = p[3]

return, Amp*exp(-0.5*(x-xc)^2./W^2.)+const
end

Pro GaussResFit

; plothist, newr, xhist, yhist, bin=.03, color=cgcolor("red"), /overplot

; yhist*=1.

; print, xhist, yhist

xhist=[3.22500, 3.25500, 3.28500, 3.31500, 3.34500, 3.37500, 3.40500, 3.43500, 3.46500,
3.49500, 3.52500, 3.55500]

yhist=[82.0000, 79.0000, 63.0000, 54.0000, 46.0000, 46.0000, 48.0000, 64.0000, 68.0000,
89.0000, 81.0000, 117.000]
```

```
plot, xhist, yhist, psym=4, color=0

st=[-71.0000,      3.27831,     0.105548,      117.000]
err=fltarr(n_elements(xhist))+0.001

name=['amp','xc','w','amp0']
np=n_elements(name)
pi =  replicate({fixed:0,limited:[0,0],limits:[0.D,0.D],MPMAXSTEP: [0.D,0.D,0.D,0.D],
parname:['']},np)
for i=0, np-1 do pi[i].parname=name[i]
fixed=fltarr(n_elements(name))
limited = [[1,1],[1,1],[1,1],[1,1]]
 limits=[[st[0]-2.,st[0]+2.],[st[1]-.2,st[1]+.2],[st[2]-.03,s t[2]+.03],[st[3]-2.,st[3]+2.]]
MPMAXSTEP=[1., 0.2, 0.05, 1.]

pi.limited=limited
pi.limits=limits
pi.fixed=fixed
pi.MPMAXSTEP=MPMAXSTEP

params = mpfitfun('gaussianfit', xhist, yhist, err, st, parinfo=pi, perror=perror)
fitg = call_function('gaussianfit', xhist, params)
oplot, xhist, fitg, color=0, thick=2.


stop
END
```

Subject: Re: Fitting a Gaussian to an "unleveled" histogram: Interpreting the width
Posted by Craig Markwardt on Sat, 14 Feb 2015 23:06:54 GMT

View Forum Message <> Reply to Message

On Wednesday, February 11, 2015 at 7:52:09 PM UTC-5, Maryam wrote:

> Hello.
>
> I am trying to fit an inverted Gaussian function to a histogram that is higher at one end than the other. I simply define a function that is a Gaussian with negative amplitude (plus a constant to shift the Gaussian up along the y-axis) and call that function using mpfitfun to get the fit. It all works out fine and I do get a fit that looks reasonable. However, I am not sure how to interpret the standard deviation returned by mpfitfun.

It really depends why the histogram is "higher at one end than the other." Is it because there is a sloping continuum or background level? If so, why not fit a constant plus linear term to the background instead of just a constant? Make your model more realistic and you will get more realistic parameter values. Make your model more synthetic, and your parameters will be meaningless.

CM

Subject: Re: Fitting a Gaussian to an "unleveled" histogram: Interpreting the width
Posted by Maryam on Fri, 27 Feb 2015 20:25:22 GMT
View Forum Message <> Reply to Message

Hello Craig,

Thanks for your reply. It is actually not an absorption line in a spectrum that I am trying to fit the Gaussian to. It is just the distribution of asteroids in the asteroid belt. I am trying to measure the width of the gap caused by the planet's (Jupiter's) gravitational perturbation; and the gap has somewhat of a Gaussian-looking shape, except that it is higher at one end than the other. IDL's mpfitfun fits it nicely but I want to know how it determines the amplitude. I tried normalizing the histogram once to the lower end and once to the higher end and then made the fit to see which amplitude it is that IDL is using, but it is neither one of those! The fit works, it is just the interpretation of the parameters that I am confused about!

Maryam