Subject: Re: Newbie's question
Posted by James Kuyper on Thu, 20 Oct 2005 18:41:08 GMT
View Forum Message <> Reply to Message

ChiChiRuiz@gmail.com wrote:

> Hi there,
>
> I have a scatter plot which has the shape of a parabola, like y=x^2.
> I want to find the best curve fit to the scatter plot, so I used the
> function "curvefit" with no weights and with initial guesses (1.0, 2.0)
> i.e. y = 1.*x^(2.). So, here's the problem...when I use only the right
> half of the data points (i.e. x and y values are positive), I get the
> curvefit returns parameter (0.5, 1.5), which means, the best fit curse
> is y=.5*x^(1.75). I know the fit should be symmetric, so the same curve
> SHOULD fit the other half. Now unto the left half side of the data
> set, curvefit does not work anymore, and here's why, x^(1.5)=x^(3/2)
> and when x is a negative number, IDL returns "NaN" because it can't
> take the square root of a negative number, hence the entire procedure
> will not work. I ended up having to throw away half of my data points,
> and I'm not very comfortable with that. Any idea how to go around it
> or suggest another function to do the same thing?

The fundamental problem is that curve fitting routines generally require that the dependent variable is a well-defined and continuous function of the curve's parameters. x^a is well-defined for negative numbers, only if it is treated as a complex-valued expression. It's continuous only if you use an unconventional branch cut, one that doesn't run along the negative real axis. If you have no idea what a branch cut is, you shouldn't even be attempting to do a fit of this type.

That's just a symptom of a deeper and simpler problem: you should try to fit data to a function, unless you have an understanding of the data that suggests that a function of that type is to be expected.

Of course, sometimes you have to fit the data without having any theoretical basis for the fit. As long as you have reason to believe that the dependent variable is a sufficiently continuous function of the independent variables, you can usually fit it to a polynomial series ("sufficiently" and "usually" are weasel words to cover many different complicated issues that would require a small book to explain them properly).

There's many different polynomial series you can fit to - the general rule is that if you use a sufficiently large number of terms to fit your data, the remaining error in the fit will be dominated by a term proportional to the first term in the series that you didn't use. For instance, in a simple power series, if you fit y = a + b*x + c*x^2, then the first term you left out is x^3, so you should expect the errors to be roughly proportional to x^3; they'll be smallest near x == 0. Similarly, if you fit to shifted power series like y = a + b*(x-x0) + c*(x-x0)^2, where x0 is fixed, then the first term you left out was (x-x0)^3. Therefore, your errors will tend to be smallest near x == x0.

> Besides, I've thought about using "polyfit", but if I remember
> correctly, polyfit only takes in one x value vs. one y value. Scatter
> plot has one x value vs. several y values. I don't think it'll
> work in my case, but I may be wrong...

POLY_FIT is a suitable routine for performing such a fit. I don't understand what you're saying about why you don't think you can use it, but your reason sounds incorrect. You normally send polyfit a complete set of x values, and a complete set of corresponding y values.

```
x = (INDGEN(32)-16)/16.0
y = (x-2.0)*x*(x+2.0); Cubic function

fit = POLY_FIT(x,y,2,YFIT=yfit); Quadratic fit
plot,x,y,psym=2
oplot,x,yfit; Fairly good fit of quadratic curve to cubic data.
plot,x,yfit-y;
```