Subject: Multidimensional curve fitting
Posted by keith on Fri, 19 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

I have a 2-dimensional dataset which I wish to parameterize with a scalar function of 2 variables in the form y=f(x1,x2) (and in the future I will extend this to higher dimensionalities). I would prefer a nonlinear function f(), but could make do with a polynomial of smallish order (<5).

The usual IDL fitting routines svdfit and curvefit only deal with 1-d functions. There is a function "sfit" which claims to perform surface fitting, but this can not provide uncertainties in the fit, nor even take account of the numerical values of x1, x2.

The JHU usr library funtion opfit2d doesn't really do what I want either. Although orthogonal polynomials are nice, I need the "ordinary" polynomial coefficients in order to taka analytical derivatives.

Do any of you have ideas/suggestions/routines which might help?

Thanks

Keith Refson
--

 ---------- ----------
| Email : keith@earth.ox.ac.uk | Dr Keith Refson, Dept of Earth Sciences|
| TEL(FAX): +44 1865 272026 (272072)| Parks Road, Oxford OX1 3PR, UK |
 ---------- ----------


Subject: Re: Multidimensional curve fitting
Posted by rivers on Fri, 19 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <1995May19.085532.18482@rahman.earth.ox.ac.uk>, keith@earth.ox.ac.uk (Keith Refson) writes:
> I have a 2-dimensional dataset which I wish to parameterize with a
> scalar function of 2 variables in the form y=f(x1,x2) (and in the
> future I will extend this to higher dimensionalities). I would prefer
> a nonlinear function f(), but could make do with a polynomial of
> smallish order (<5).
>
> The usual IDL fitting routines svdfit and curvefit only deal with 1-d
> functions. There is a function "sfit" which claims to perform surface
> fitting, but this can not provide uncertainties in the fit, nor even
> take account of the numerical values of x1, x2.

I don't think it is true that CURVEFIT can only deal with 1-d functions.
CURVEFIT optimizes parameters to minimize the sum of the squares of the
differences between an observed data set and a predicted data set. The
independent variable, dependent variable and predictions are must be passed
as 1-D vectors, but there is no restriction on the number of dimensions the data
really represent. CURVEFIT has no trouble fitting a 2-D data set
if you REBIN the arrays to 1-D before passing them.


Mark Rivers (312) 702-2279 (office)
CARS (312) 702-9951 (secretary)
Univ. of Chicago (312) 702-5454 (FAX)
5640 S. Ellis Ave. (708) 922-0499 (home)
Chicago, IL 60637 rivers@cars3.uchicago.edu (Internet)


---

Subject: Re: Multidimensional curve fitting
Posted by rivers on Sat, 20 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <1995May19.085532.18482@rahman.earth.ox.ac.uk>, keith@earth.ox.ac.uk (Keith Refson) writes:
> The usual IDL fitting routines svdfit and curvefit only deal with 1-d
> functions. There is a function "sfit" which claims to perform surface
> fitting, but this can not provide uncertainties in the fit, nor even
> take account of the numerical values of x1, x2.
>
Here is an example which illustrates my previous post. It computes and fits a
2-D surface. It is necessary to "REFORM" the array to 1-D before passing it to
CURVEFIT, but this is only a minor nuisance. This program uses the new version
of CURVEFIT which does not require derivatives, but the old version will work
the same in terms of fitting N-dimensional data.


Mark Rivers (312) 702-2279 (office)
CARS (312) 702-9951 (secretary)
Univ. of Chicago (312) 702-5454 (FAX)
5640 S. Ellis Ave. (708) 922-0499 (home)
Chicago, IL 60637 rivers@cars3.uchicago.edu (Internet)


```
 ************************************************************
pro fit_curve, ind, g, pred
nx = 10
ny = 8
x = rebin(findgen(nx), nx, ny)
y = rebin(transpose(findgen(ny)), nx, ny)
pred = g(0)*(x-g(1))^2 + g(2)*(y-g(3))^2    ; Predicted surface based on
```

```
                                    ; current fit parameters
pred = reform(pred, nx*ny, /overwrite)      ; Convert back to 1-D array
end


; Main program - compute a 2-D surface and fit it
nx = 10
ny = 8
x = rebin(findgen(nx), nx, ny)
y = rebin(transpose(findgen(ny)), nx, ny)
a = [4.5, 4.7, 6, 5]                        ; Actual coefficients
g = [4.0, 3.0, 5.4, 5.9]                    ; Guess of coefficients
obs = a(0)*(x-a(1))^2 + a(2)*(y-a(3))^2     ; 2-D surface
obs = reform(obs, nx*ny, /overwrite)        ; Reform to 1-D for CURVEFIT
w = fltarr(nx*ny) + 1.                      ; Weights, all 1
ind = findgen(nx*ny)                        ; Independent variable, dummy
print, 'Actual coefficients   = ', a
print, 'Initial guess         = ', g
fit = curvefit(ind, obs, w, g, funct='fit_curve', /noderivative)
print, 'Best fit coefficients = ', g
end
```

---

Subject: Re: Multidimensional curve fitting
Posted by cpylant on Wed, 24 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <1995May19.085532.18482@rahman.earth.ox.ac.uk>, keith@earth.ox.ac.uk (Keith Refson) says:
>
> The usual IDL fitting routines svdfit and curvefit only deal with 1-d
> functions. There is a function "sfit" which claims to perform surface
> fitting, but this can not provide uncertainties in the fit, nor even
> take account of the numerical values of x1, x2.
>
You might look at the curve-fitting section of "Numerical Recipes in C". The code they present is seemingly one dimensional, but they demonstrate how it can easily be 'tricked' into fitting to multiple dimensions. I don't know if the IDL routines could be similarly tricked.

Good luck,

Chris Pylant

---