
Subject: Re: multi-dimensional SDVFIT

Posted by [R.G.S.](#) on Tue, 03 Jul 2001 21:17:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

Ingo Wardinski <ingo@gfz-potsdam.de> wrote in message
news:ynlelryc66q.fsf@mt35.gfz-potsdam.de...

> Hi,
> does anyone know, if it is possible to make a multi-dimensional SDVFIT
> to a certain dataset? I'm looking for a SDVFIT of $y=f(x_1, x_2)$, but
> AFAIK the standard idl SDVFIT routine works only for 1-d.
> thanks in advance
> ingo
> --
> ingo wardinski ingo@gfz-potsdam.de
> GeoForschungsZentrum Potsdam, Telegrafenberg F456, 14473 Potsdam
>
> "Ich war ganz allein in der Maschine und habe dem Piloten gesagt..."
> (Helmut Kohl in "Am Ende des Jahrhunderts")

Hello,
the SVD solution to a least squares fit ($Ax=b$) to higher dimensions is
trivial.
Merely add on the other dimensions as columns of A, since there is no
difference
in principal between " x^2 " and " xy ", since to the SVD, the matrix A is
merely a collection
of numbers.

Cheers,
bob stockwell
stocwkell at co-ra dot com

For instance, here is a smallpiece of code that will compute a fit to
 $f(x,y,z) = a_0 + a_1x + a_2y + a_3z + a_4xy + a_5xz + a_6yz + a_7x^2 + a_8y^2 + a_9z^2$

(i.e. 10 parameters second order term polynomial in 3D)

```
; Here zon = data(x,y,z)
m = 10 ; number of terms in equation
n = n_elements(zon) ; n = number of data points
```

```
; create matrix column by column
a = dblarr(m,n)
a(0,*) = 1
a(1,*) = x
a(2,*) = y
a(3,*) = z
a(4,*) = x*y
a(5,*) = x*z
a(6,*) = y*z
a(7,*) = x^2
a(8,*) = y^2
a(9,*) = z^2
```

```
; Decompose A:
tic = systime(1)
SVDC, A, W, U, V,/double
toc = systime(1)
```

```
; Compute the solution and print the result:
result1 = SVSOL(U, W, V, zon,/double)
toc2 = systime(1)
print,result1
```
