Subject: SVDFIT bug?
Posted by justin_ashmall on Mon, 13 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Greetings all,
I came across what looks like a bug in SVDFIT. I submitted it to RSI a week ago now but haven't heard back (except for a confirmation of receipt) and thought it might of interest to The Group. The bug appears to be in the errors given for the coefficients returned by SVDFIT. Below is the email I sent to RSI explaining the problem. I'm using IDL5.3 on an NT box.

Justin

I've used LINFIT to fit a straight line to some data, and done the same using SVDFIT (with M set to 2 for a linear fit). Whilst I get exactly the same coefficients returned from both routines, the standard deviations (errors) of the coefficients vary between routines. It looks to me that the SVDFIT errors are incorrect.

Strangely it seems that the errors of the coefficients from SVDFIT do not depend on the y values. I've included a short prog to demonstrate the problem. The results it produces are shown below:

```
IDL> lin_test
Fitting y = a + bx
Showing: a, b, a_err, b_err

Fitting (x, y1):
SVDFIT:    -0.084282358      0.14992642      0.42803180     0.010803003
LINFIT:    -0.084282358      0.14992642     0.085521095    0.0021584486
Fitting (x, y2)
SVDFIT:      0.15193113       3.5026045      0.42803180     0.010803003
LINFIT:      0.15193113       3.5026045      0.58222837     0.014694737
```

You can see that the last two numbers (the errors) vary between LINFIT and SVDFIT. Notice also that the error values from SVDFIT are the same with two different sets of y values (but the same x values).

From the documentation we see that the SIGMA keyword returns a "vector of standard deviations for the returned coefficients" with SVDFIT. With LINFIT the SIGMA keyword returns a "vector of probable uncertainties for the model parameters." Given this maybe we should not expect the values to be the same, however order of magnitude differences and the independece of

the y values suggest something is amiss. Also, since both routines appear to use code lifted or translated from Numerical Recipes, we might expect the same values back.

```
PRO lin_test

 ;Create some data
 x =DOUBLE([85,  76,  24,  21,  8.6, 5.7, 1.6, 1.2, 0.6 ])
 y1=DOUBLE([13,  11,  3.3, 3.0, 1.3, 0.8, 0.2, 0.1, 0.08])
 y2=DOUBLE([296, 268, 84,  76,  30,  19,  5.6, 4.3, 2.0 ])

 PRINT, "Fitting y = a + bx"
 PRINT, "Showing: a, b, a_err, b_err"
 PRINT

 PRINT,"Fitting (x, y1):"
 ;Make linear fit using SVDFIT, setting M = 2
 svd_vals1 = SVDFIT( x, y1, 2, /DOUBLE, SIGMA=svd_sig1)
 PRINT, "SVDFIT:", svd_vals1[0], svd_vals1[1], svd_sig1[0], svd_sig1[1]

 ;Fit same data using LINFIT
 lin_vals1 = LINFIT( x, y1, /DOUBLE, SIGMA=lin_sig1)
 PRINT, "LINFIT:", lin_vals1[0], lin_vals1[1], lin_sig1[0], lin_sig1[1]


 PRINT, "Fitting (x, y2)"
 ;SAme as above with y2 data
 svd_vals2 = SVDFIT( x, y2, 2, /DOUBLE, SIGMA=svd_sig2)
 PRINT, "SVDFIT:", svd_vals2[0], svd_vals2[1], svd_sig2[0], svd_sig2[1]

 ;Fit y2 data using LINFIT
 lin_vals2 = LINFIT( x, y2, /DOUBLE, SIGMA=lin_sig2)
 PRINT, "LINFIT:", lin_vals2[0], lin_vals2[1], lin_sig2[0], lin_sig2[1]


END
```

Subject: Re: SVDFIT bug?
Posted by David McClain on Tue, 14 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

PS: The same indpendence from the y values is true for ANY least squares fit, or even for the case of a completely determined system. The covariance matrix for a system defined by

$$A\,x = y$$

is always independent of the y values. It simply measures the
appropriateness of a chosen set of basis functions, not how well they fit
the data... That goodness of fit is measured by the ChiSquare estimate!

- DM

Justin <justin_ashmall@hotmail.com> wrote in message
news:8EF66B4F8ltbyltbmltbouts@155.198.199.181...
> Greetings all,
> I came across what looks like a bug in SVDFIT. I submitted it to RSI a
week
> ago now but haven't heard back (except for a confirmation of receipt) and
> thought it might of interest to The Group. The bug appears to be in the
> errors given for the coefficients returned by SVDFIT. Below is the email I
> sent to RSI explaining the problem. I'm using IDL5.3 on an NT box.
>
> Justin
>
>
>
>
> I've used LINFIT to fit a straight line to some data, and done the same
> using SVDFIT (with M set to 2 for a linear fit). Whilst I get exactly the
> same coefficients returned from both routines, the standard deviations
> (errors) of the coefficients vary between routines. It looks to me that
> the SVDFIT errors are incorrect.
>
> Strangely it seems that the errors of the coefficients from SVDFIT do not
> depend on the y values. I've included a short prog to demonstrate the
> problem. The results it produces are shown below:
>
>
> IDL> lin_test
> Fitting y = a + bx
> Showing: a, b, a_err, b_err
>
> Fitting (x, y1):
> SVDFIT: -0.084282358 0.14992642 0.42803180 0.010803003
> LINFIT: -0.084282358 0.14992642 0.085521095 0.0021584486
> Fitting (x, y2)
> SVDFIT: 0.15193113 3.5026045 0.42803180 0.010803003
> LINFIT: 0.15193113 3.5026045 0.58222837 0.014694737
>
>
> You can see that the last two numbers (the errors) vary between LINFIT and
> SVDFIT. Notice also that the error values from SVDFIT are the same with
> two different sets of y values (but the same x values).

>
> From the documentation we see that the SIGMA keyword returns a "vector of
> standard deviations for the returned coefficients" with SVDFIT. With
> LINFIT the SIGMA keyword returns a "vector of probable uncertainties for
> the model parameters." Given this maybe we should not expect the values to
> be the same, however order of magnitude differences and the independece of
> the y values suggest something is amiss. Also, since both routines appear
> to use code lifted or translated from Numerical Recipes, we might expect
> the same values back.
>
>
> PRO lin_test
>
> ;Create some data
> x =DOUBLE([85, 76, 24, 21, 8.6, 5.7, 1.6, 1.2, 0.6])
> y1=DOUBLE([13, 11, 3.3, 3.0, 1.3, 0.8, 0.2, 0.1, 0.08])
> y2=DOUBLE([296, 268, 84, 76, 30, 19, 5.6, 4.3, 2.0])
>
> PRINT, "Fitting y = a + bx"
> PRINT, "Showing: a, b, a_err, b_err"
> PRINT
>
> PRINT,"Fitting (x, y1):"
> ;Make linear fit using SVDFIT, setting M = 2
> svd_vals1 = SVDFIT(x, y1, 2, /DOUBLE, SIGMA=svd_sig1)
> PRINT, "SVDFIT:", svd_vals1[0], svd_vals1[1], svd_sig1[0], svd_sig1[1]
>
> ;Fit same data using LINFIT
> lin_vals1 = LINFIT(x, y1, /DOUBLE, SIGMA=lin_sig1)
> PRINT, "LINFIT:", lin_vals1[0], lin_vals1[1], lin_sig1[0], lin_sig1[1]
>
>
> PRINT, "Fitting (x, y2)"
> ;SAme as above with y2 data
> svd_vals2 = SVDFIT(x, y2, 2, /DOUBLE, SIGMA=svd_sig2)
> PRINT, "SVDFIT:", svd_vals2[0], svd_vals2[1], svd_sig2[0], svd_sig2[1]
>
> ;Fit y2 data using LINFIT
> lin_vals2 = LINFIT(x, y2, /DOUBLE, SIGMA=lin_sig2)
> PRINT, "LINFIT:", lin_vals2[0], lin_vals2[1], lin_sig2[0], lin_sig2[1]
>
>
> END
>
>

Subject: Re: SVDFIT bug?
Posted by David McClain on Tue, 14 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Indeed, the covariance matrix for an SVDFIT depends only upon the values of the independent variable and the weight accorded to each measurement...

You can see this for yourself if you remember that the SVD decomposition is defined as

A = U W (trn V)

where matrix A is decomposed into a row-orthogonal matrix U, a diagonal matrix W, and another row-orthogonal matrix V. The covariance matrix is obtained as

cov = ((inv W) #* V) #* (trn (inv W) #* V)

Since this does not depend on the dependent data values, the covariance is seen to depend only on the independent values and their associated weights.

- DM

Justin <justin_ashmall@hotmail.com> wrote in message news:8EF66B4F8ltbyltbmltbouts@155.198.199.181...
> Greetings all,
> I came across what looks like a bug in SVDFIT. I submitted it to RSI a week
> ago now but haven't heard back (except for a confirmation of receipt) and
> thought it might of interest to The Group. The bug appears to be in the
> errors given for the coefficients returned by SVDFIT. Below is the email I
> sent to RSI explaining the problem. I'm using IDL5.3 on an NT box.
>
> Justin
>
>
>
>
> I've used LINFIT to fit a straight line to some data, and done the same
> using SVDFIT (with M set to 2 for a linear fit). Whilst I get exactly the
> same coefficients returned from both routines, the standard deviations
> (errors) of the coefficients vary between routines. It looks to me that
> the SVDFIT errors are incorrect.
>
> Strangely it seems that the errors of the coefficients from SVDFIT do not
> depend on the y values. I've included a short prog to demonstrate the
> problem. The results it produces are shown below:
>
>

> IDL> lin_test
> Fitting y = a + bx
> Showing: a, b, a_err, b_err
>
> Fitting (x, y1):
> SVDFIT: -0.084282358 0.14992642 0.42803180 0.010803003
> LINFIT: -0.084282358 0.14992642 0.085521095 0.0021584486
> Fitting (x, y2)
> SVDFIT: 0.15193113 3.5026045 0.42803180 0.010803003
> LINFIT: 0.15193113 3.5026045 0.58222837 0.014694737
>
>
> You can see that the last two numbers (the errors) vary between LINFIT and
> SVDFIT. Notice also that the error values from SVDFIT are the same with
> two different sets of y values (but the same x values).
>
> From the documentation we see that the SIGMA keyword returns a "vector of
> standard deviations for the returned coefficients" with SVDFIT. With
> LINFIT the SIGMA keyword returns a "vector of probable uncertainties for
> the model parameters." Given this maybe we should not expect the values to
> be the same, however order of magnitude differences and the independece of
> the y values suggest something is amiss. Also, since both routines appear
> to use code lifted or translated from Numerical Recipes, we might expect
> the same values back.
>
>
> PRO lin_test
>
> ;Create some data
> x =DOUBLE([85, 76, 24, 21, 8.6, 5.7, 1.6, 1.2, 0.6])
> y1=DOUBLE([13, 11, 3.3, 3.0, 1.3, 0.8, 0.2, 0.1, 0.08])
> y2=DOUBLE([296, 268, 84, 76, 30, 19, 5.6, 4.3, 2.0])
>
> PRINT, "Fitting y = a + bx"
> PRINT, "Showing: a, b, a_err, b_err"
> PRINT
>
> PRINT,"Fitting (x, y1):"
> ;Make linear fit using SVDFIT, setting M = 2
> svd_vals1 = SVDFIT(x, y1, 2, /DOUBLE, SIGMA=svd_sig1)
> PRINT, "SVDFIT:", svd_vals1[0], svd_vals1[1], svd_sig1[0], svd_sig1[1]
>
> ;Fit same data using LINFIT
> lin_vals1 = LINFIT(x, y1, /DOUBLE, SIGMA=lin_sig1)
> PRINT, "LINFIT:", lin_vals1[0], lin_vals1[1], lin_sig1[0], lin_sig1[1]
>
>
> PRINT, "Fitting (x, y2)"

```
>  ;SAme as above with y2 data
>  svd_vals2 = SVDFIT( x, y2, 2, /DOUBLE, SIGMA=svd_sig2)
>  PRINT, "SVDFIT:", svd_vals2[0], svd_vals2[1], svd_sig2[0], svd_sig2[1]
>
>  ;Fit y2 data using LINFIT
>  lin_vals2 = LINFIT( x, y2, /DOUBLE, SIGMA=lin_sig2)
>  PRINT, "LINFIT:", lin_vals2[0], lin_vals2[1], lin_sig2[0], lin_sig2[1]
>
>
>  END
>
>
```