Subject: SVDFIT Problems
Posted by William Clodius on Fri, 30 Aug 2002 16:06:11 GMT
View Forum Message <> Reply to Message

In my occaisional attempts to use SVDFIT I have had problems in that it reports singular solutions when other fitting routines, e.g., REGRESS, have had not problems. The obvious source of the problems is that within the routine the line

```
    THRESH = 1.0E-9
```

does not correspond to the definition in Numerical Recipes

```
    THRESH = TOL*wmax
```

THRESH is used to determine the number of singular values.

The definition in Numerical Recipes has two advantages

1. TOL can be changed depending on whether the routines are single or double precision

2. wmax provides a natural scaling for the problem.

Therefore the code has two problems

1. The determination of the number of singular values is effectively independent of the double keyword, although that determination should depend on the relative tolerance of the math

2. The lack of a natural scale factor means that the reporting of singularity depends on the physical units the equations represent, e.g., singularity determination for an equation involving lengths, will depend on whether the lengths are reported in bohr, angstroms, microns, meters, kilometers etc. A properly implemented SVD rescales during its calculations so that unless an underflow or overflow occurs in computing the rescaling the presence of singularities is independent of the units.

The second problem can be reduced by choosing units (or a data rescaling) so the independent variable and the values returned by FUNCTION_NAME are typically of magnitude one, but even then the potential remains that singular values are over or under reported.

Unfortunately it appeas that the code currently relies on a simplified version of the Numerical Recipes routine SVDCMP, that does not return the singular value vector that Numerical Recipes uses to compute wmax. A rewrite to give SVDFIT the proper properties is more work than I am willing to put in at this time.

Subject: Re: SVDFIT Problems
Posted by Chris Torrence on Tue, 03 Sep 2002 15:20:39 GMT
View Forum Message <> Reply to Message

Hi Bill,

Actually, the SVDFIT code *does* use THRESH = TOL*wmax. If you look
carefully at the IDL code, around line 249, there is the following line:

```
        small=WHERE(variance LE max(variance)*thresh, cc)
```

(The variable name "thresh" was an unfortunate choice, and should really
have been "tol".)

Internally, the C code is identical to the Numerical Recipes code, except
for the TOL value, which is 1e-9 for both single and double precision. We
could consider adding a TOL keyword to the SVDFIT function, which would
allow the user to change this default.

Cheers,

Chris
Research Systems, Inc.

"Bill" <wclodius@lanl.gov> wrote in message
news:3D6F97ED.C81C6C00@lanl.gov...
> In my occaisional attempts to use SVDFIT I have had problems in that it
> reports singular solutions when other fitting routines, e.g., REGRESS,
> have had not problems. The obvious source of the problems is that within
> the routine the line
>
>   THRESH = 1.0E-9
>
> does not correspond to the definition in Numerical Recipes
>
>   THRESH = TOL*wmax
>

Subject: Re: SVDFIT Problems
Posted by William Clodius on Thu, 05 Sep 2002 17:26:26 GMT
View Forum Message <> Reply to Message

Chris Torrence wrote:

> Hi Bill,
>
> Actually, the SVDFIT code *does* use THRESH = TOL*wmax. If you look

> carefully at the IDL code, around line 249, there is the following line:
>
>   small=WHERE(variance LE max(variance)*thresh, cc)
>
> (The variable name "thresh" was an unfortunate choice, and should really
> have been "tol".)
>
> Internally, the C code is identical to the Numerical Recipes code, except
> for the TOL value, which is 1e-9 for both single and double precision. We
> could consider adding a TOL keyword to the SVDFIT function, which would
> allow the user to change this default.
>
> Cheers,
>
> Chris
> Research Systems, Inc.
> <snip>

Two comments:

1. Unless the double keyword is being ignored, TOL should not be the same for single and double precision. All computers IDL is currently available on use IEEE 754 math. In this standard the mantissa is represented by 23 bits in single precision and 52 bits in double precision. With 754's hidden bit, single precision has a relative precison of 1/2^24 ~ 6e-8 and double has a relative precision of 1/2^53 ~ 1e-16. SVD should identify as singular any value that is largely determined by the precision of the arithmetic. Such values will be less than a small multiple of the relative precision with the largest eigenvalue. For single precision the Numerical Recipes code uses a value of 1e-5, or about 150 times the precision. The current value of 1.e-9 will miss many values that are effectively singular for single precision calculations. If double precision is used on more complicated problems, then perhaps a reasonable estimate of TOL is 150^2 * 1e-16 ~ 2e-12. The current value of 1.e-9 will treat many values that are not singular as singular for double precision calculations.

2. Even given the above I strongly believe that either Numerical Recipe's implementation of SVDFIT is wrong, or IDL has made a mistake in implementing it. For infinite precision arithmetic, scaling for each sample "measurement" the dependent variable, the vector of values returned by the function of the measurement, and the dependent variable's sigma by the same sample dependent finite value has no effect on the solution of the equations. While that does not strictly hold for finite precision arithmetic, I would expect it to almost always hold for relative changes less than the square root of the precision. I would also expe that using such scalings to ensure that the typical values of the functions are on the order of one would improve the regression. Unfortunately doing this on my fits to proprietary data increases the number of sinularities identified by SVDFIT.

Subject: Re: SVDFIT Problems
Posted by James Kuyper on Thu, 05 Sep 2002 20:15:07 GMT
View Forum Message <> Reply to Message

Bill wrote:
>
> Chris Torrence wrote:
>
>> Hi Bill,
>>
>> Actually, the SVDFIT code *does* use THRESH = TOL*wmax. If you look
>> carefully at the IDL code, around line 249, there is the following line:
>>
>>   small=WHERE(variance LE max(variance)*thresh, cc)
>>
>> (The variable name "thresh" was an unfortunate choice, and should really
>> have been "tol".)
>>
>> Internally, the C code is identical to the Numerical Recipes code, except
>> for the TOL value, which is 1e-9 for both single and double precision. We
>> could consider adding a TOL keyword to the SVDFIT function, which would
>> allow the user to change this default.
>>
>> Cheers,
>>
>> Chris
>> Research Systems, Inc.
>> <snip>
>
> Two comments:
>
> 1. Unless the double keyword is being ignored, TOL should not be the same for
> single and double precision. All computers IDL is currently available on use
> IEEE 754 math. In this standard the mantissa is represented by 23 bits in
> single precision and 52 bits in double precision. With 754's hidden bit,
> single precision has a relative precison of 1/2^24 ~ 6e-8 and double has a
> relative precision of 1/2^53 ~ 1e-16. SVD should identify as singular any
> value that is largely determined by the precision of the arithmetic. Such

True, but it should also identify as singular any value that is largely
determined by the precision of the input data. With double precision
floating point, the precision of the result is likely to be dominated by
the precision of the input data, not by the precision of the arithmetic.

That's why TOL shouldn't just slavishly depend upon the precision of the
data type. It's also why TOL should be adjustable by the user.

Subject: Re: SVDFIT Problems
Posted by William Clodius on Thu, 05 Sep 2002 23:43:20 GMT
View Forum Message <> Reply to Message

James Kuyper wrote:

> Bill wrote:
>> <snip>
>>
>> Two comments:
>>
>> 1. Unless the double keyword is being ignored, TOL should not be the same for
>> single and double precision. All computers IDL is currently available on use
>> IEEE 754 math. In this standard the mantissa is represented by 23 bits in
>> single precision and 52 bits in double precision. With 754's hidden bit,
>> single precision has a relative precison of 1/2^24 ~ 6e-8 and double has a
>> relative precision of 1/2^53 ~ 1e-16. SVD should identify as singular any
>> value that is largely determined by the precision of the arithmetic. Such
>
> True, but it should also identify as singular any value that is largely
> determined by the precision of the input data. With double precision
> floating point, the precision of the result is likely to be dominated by
> the precision of the input data, not by the precision of the arithmetic.
>
> That's why TOL shouldn't just slavishly depend upon the precision of the
> data type. It's also why TOL should be adjustable by the user.

Not quite true. TOL in the original Numerical Recipes routine is intended to
represent exclusively the effects of the machine numerics on the solution of the
equations. The effects of the precision of the input data are in effect handled by
the keyword MEASURE_ERRORS in what is in the current version of IDL's SVDFIT.
They need to be separated because in linear regression the effects of numerical
precision tends to increase with an increasing number of measurements included in
the regression, while effect of measurement precision (provided the measurements
are independent) tends to decrease with an increasing number of measurements.

This discussion then promted me to look more closely at IDL's code. In that code
the determination of singularity uses the condition
   small = where(VARIANCE lt max(VARIANCE)*THRESH, count)
where apparently THRESH is intended to replace Numerical Recipe's TOL. However in
my old Fortran version of Numerical Recipes, (I don't have a C version readilly
available), the conddition is acctually equivalent to
   small = where(SINGULAR_VALUE lt max(SINGULAR_VALUE)*TOL, count)
While the variance is related to the singular value through, the V matrix they are
not linearly related (the variance is linearly related to the INVERSE SQUARES of
the singular values) and there is no reason to expect that the two conditions
would be equivalent. The call to NR__SVDFIT that does the actual fitting in
SVDFIT does not return the singular value matrix or the V matrix so I cannot
currently do a detailed analysis, but I suspect the comparison is incorrect.

Subject: Re: SVDFIT Problems
Posted by William Clodius on Fri, 06 Sep 2002 17:08:25 GMT
View Forum Message <> Reply to Message

Bill wrote:<snip>

```
>
>  This discussion then promted me to look more closely at IDL's code.  In that code
>  the determination of singularity uses the condition
>      small = where(VARIANCE lt max(VARIANCE)*THRESH, count)
>  where apparently THRESH is intended to replace Numerical Recipe's TOL.  However in
>  my old Fortran version of Numerical Recipes, (I don't have a C version readilly
>  available), the conddition is acctually equivalent to
>      small = where(SINGULAR_VALUE lt max(SINGULAR_VALUE)*TOL, count)
>  While the variance is related to the singular value through, the V matrix they are
>  not linearly related (the variance is linearly related to the INVERSE SQUARES of
>  the singular values) and there is no reason to expect that the two conditions
>  would be equivalent.   The call to NR__SVDFIT that does the actual fitting in
>  SVDFIT does not return the singular value matrix or the V matrix so I cannot
>  currently do a detailed analysis, but I suspect the comparison is incorrect.
```

As I seem to be in a posting mood note that

```
    small = where(VARIANCE lt max(VARIANCE)*THRESH, count)
```

identifes as small those quantities with the smallest magnitude variances. If relative magnitude (i.e., units) can be ignored, the current code reports as singular precisely those quantities that are best determined given the uncertainties in the input data, while singular is intended to identify those quantities that may be badly determined due to numerical problems.

Subject: Re: SVDFIT Problems
Posted by William Clodius on Tue, 10 Sep 2002 22:42:43 GMT
View Forum Message <> Reply to Message

Bill wrote:<snip>

```
>      small = where(VARIANCE lt max(VARIANCE)*THRESH, count)
>  identifes as small those quantities with the smallest magnitude variances.  If
>  relative magnitude (i.e., units) can be ignored, the current code reports as singular
>  precisely those quantities that are best determined given the uncertainties in the
>  input data, while singular is intended to identify those quantities that may be badly
>  determined due to numerical problems.
```

FWIW IDL support has decidet that the above comparison is indeed invalid and will be fixed in the next release of IDL.