Re: linalg[leastsqrs] in Maple V R4

Source: http://sci.tech−archive.net/Archive/sci.math.symbolic/2006−08/msg00064.html

- *From*: "lcw1964" <leslie.wright@xxxxxxxxxxxxx>
- *Date*: 30 Jul 2006 19:08:33 −0700

I have decided to write my own leastsqrs routine using linalg(Svd) which seems to be a bit more consistent. This is how I have tested it out and have convinced myself it is adequate for my own purposes.

I execute dvec:=evalf(linalg[Svd](u,Ubig,V)). This returns a vector of the singular values, all of which are nonzero and not too small. To execute the least squares estimate as described well in section 2.6 of Numerical Recipes, I create the orthonormal "left" matrix U:=evalm(delcols(Ubig, 10..72)), and the diagonal matrix of inverses of the singular values by isingmat:=diag(seq(1/dvec[i],i=1..9). The right matrix V is just what I need and can use it without modification. given this, the least squares solution is

```
sol:=evalm(V&*isingmat&*transpose(U));
```

I get the exact same result every time when I run everything after a restart, and the consistency is reassuring. The results are pretty close, at least in the first few digits, to the linalg[leastsqrs] output, though the consistency of the latter is not reassuring.

As a test, I attempt to reconstruct the original matrix u from my results,and compute its difference from the original:

```
evalm(U&*diag(seq(dvec[i],i=1..9))&*transpose(V);
evalm("−u);
```

The output here reveals differences indicating that at most 2 or 3 SDs were lost. With Digits:=60, for example, this 72x9 matrix of differences has elements ranging between about 1E−57 down to zero. For the stuff I am doing, this is tolerable and understandable.

I have no easy way of proving that the least squares result generated by this approach is "truer" or "better" than that given by the built−in leastsqrs routine−−according to the theory it is supposed to be the "right" answer−−but I do seem to think that it is at least more

consistent, and I will infer more accuracy and precision since the matrices and vectors generated by the Svd process that are prerequisites for the least squares calculation seem to maintain reasonable precision, as one can infer from the fairly close agreement when one reconstructs the original matrix from its decomposed parts.

This is an intermediate step in a larger algorithm (please refer to section 5.13 of Numerical Recipes if you are curious what I am trying to port into Maple), and I will let you know how I make out.

Les

lcw1964 wrote:

> Thanks Prof. Israel.
>
> I won't burden the group with a message filled with hundreds of decimal digits. I will do some more tests and get back to you. In the meantime, I do hav a work around−−I set digits rather high. In my particular case, I am estimating the coefficients in a rational approximation problem, so I need a good degree of decimal precision in my results.
>
> Here is an abridgement of what I am getting now. u is a 72 by 9 matrix, bb is a 72 element vector:
>
> ```
> Digits: = 50:
> <snip out lots of other stuff>
> linalg[leastsqrs](u,bb):
> linalg[leastsqrs](u,bb):
> "−"";
> ```
>
> The last command produces a 9 element vector of differences ranging between about 4E−42 to 1E−45. That is what amazes me−−why would the leastsqrs routine differ in the last 5 to 9 significant digits in two CONSECUTIVE calculations of the exact same thing? And which one, if either, is "right"? When the matrix is bigger (136 x 17), the exact same thing with Digits at 50 produces a difference vector with 17 elements ranging between 3E−25 and 6E−37, so there is discrepancy in the last 13 to 25 digits at least in consecutive calculations. Are there many potential solutions that differ only in the last several decimal digits, and Maple converges to one or the other in consecutive performances of the exact same computation? That strikes me as somewhat random.
>
> Since I know this happens, I will just be sure to set Digits to a lot more than what I need. On my machine Maple seems very fast for problems of this size so this is no hardship. In the meantime, I will see if

Svd() and manual computation of leastsquares solutions yields more reproducible results.

In the meantime, I am grateful for ongoing comments and wisdom.

Les

Robert Israel wrote:

> In article
> <1154264326.861799.211040@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
> lcw1964 <leslie.wright@xxxxxxxxxxxxx> wrote:
>
> > I know that I am using an admittedly ancient version of the software,
> > but it suits my modest personal purposes for the most part.
> >
> > I am wondering if anyone has has any experience with the leastsqrs
> > procedure, in either older or newer incarnations, only to discover that
> > it seems painfully inaccurate on repeat application.
> >
> > I have tried to use linalg[leastsqrs] to find the least squares
> > solution of an fairly hefty sized overdetermined system, about 80
> > equatons in 16 unknowns. Repeat application on the exact same
> > matrix/vector input combination yields different results with the loss
> > of many if not most of the significant digits. I am not talking just a
> > few digits lost to accumulated rounding error, I am talking a couple of
> > dozen digits when Digits is set pretty high. The problem gets worse the
> > larger the linear system is, and using the 'optimize' argument seems to
> > slow things down and doesn' add any value.
> >
> > This is driving me wonky. Is this a bug or a feature? Is it present in
> > newer versions of Maple? Would I be better off using Svd() and
> > computing the least squares solution myself from that output? I really
> > don't mind "black box" routines if they give reproducible results, and
> > linalg[leastsqrs] in my hands does not.

The LinearAlgebra package in more recent versions of Maple should be quite a bit better than linalg at most number−crunching applications. But I didn't see any significant difference between linalg[leastsqrs] and LinearAlgebra[LeastSquares] on a random 80 x 16 problem. Could you provide a specific example of the problem you're experiencing?

Robert Israel israel@xxxxxxxxxxx
Department of Mathematics http://www.math.ubc.ca/~israel
University of British Columbia Vancouver, BC, Canada

.