Subject: Re: your (re-)submitted review paper:-3
From: Spencer Graves <spencer.graves@prodsyse.com>
Date: 2023-05-17, 21:11
To: J C Nash <profjcnash@gmail.com>

p.s. Please excuse a minor revision of the previous email on this.

Hi, John:

Two more things I intended to put in that email, then forgot:

1. Could you put a pithy summary in the first couple of pages including a table of your main results, so a casual reader could get 80 percent of your message from skimming that intro? Maybe list the major package::functions you consider with columns for "pros" and "cons" or something like that?

2. Do you have a companion script file, so a user can walk through it line by line and replicate your results? In other words, turn this into a vignette.

I think these two things could help make the document much more user friendly.

Thanks,
Spencer

On 5/17/23 6:50 PM, J C Nash wrote:

> Thanks for reading more carefully that we did!
>
> I'll have to submit some corrections. Truthfully, we've been battling how to express some of these rather awkward ideas, and it took us quite a bit of work to tease some of them out. The nls class objects are really Byzantine.
>
> I've been banging my head against the issue of bounds constraints and uncertainty measures for a few decades. Hence a philosophical matter. I agree there are other methods, but we were pushing hard to keep length of paper down. It's already complicated enough, of course.
>
> Best, JN
>
> On 2023-05-17 18:38, Spencer Graves wrote:
>
> > Hi, John:
> >
> > I have a quibble with the second expression for H[i,j] as the sum of two sums, the first being J'J and the second the sum over k of the partial of r[k] with respect to p[i] and p[j]: Don't you need a factor of 2 in front of the sum of those two sums?
> >
> > Also, to motivate "Newton's equation", might it make sense to define g = 2*J'r as "The gradient of S(p)" in the fourth equation above?
> >
> > Another quibble: On p. 9, I read, "To actually access the data from this element, we need to use they syntax: "eval(parse(text=result$data))". Questions about this:
> >
> > 1. I had to read the preceding paragraph multiple times to see "result" defined. How about dropping "the" from "Let us use the result as the returned object from ...", so it reads, "Let 'result' be the object returned from ...".

2. Does "eval(parse(text=result$data))", no matter whether "result" is returned from nls(), minpack.lm;;nlsLM(), or gslnls::gal_nls()?

3. Next, I read, "if the call is made with model=TRUE, then ... we can list its contents using ls(result$model)." Have you considered writing a wrapper function (with a name like "Model") that would return the information desired from that even if result$model were NULL? (This may be a bad idea, but I felt a need to ask. I could ask the same question about "xdata".)

P. 12, "Philosophical considerations: Bounds on parameters raise some interesting questions ... . Hence, nlsr::nlxb() does not compute standard errors nor their derived statistics when bounds are active." QUESTION: What about using profile likelihood for computing likelihood ratio confidence intervals? That kind of thing is discussed in Pinheiro and Bates (2000) Mixed-effects models in S and S-Plus (Springer, pp. 85-90). I would expect that more recent research and discussions of that should be available. However, I'm not familiar with any, though I would also expect that both you and Doug might be. Also, the F distribution does have a standard deviation, and it clearly has a lower bound of 0. It would be nice, therefore, I think, if nlsr::xlxb() DID compute some kind of standard errors and derived statistics when bounds were active -- or close. For example, are there standard Monte Carlo / MCMC procedures that are recommended for such situations? I'm not current on that literature, but someone should be. There's a huge difference between a truncated standard normal above -2 and above +2 😉 If that has not been done for something like this, it could be worth doing. Obviously, I'm not talking here about what should be in this paper but rather something I'd like to see. If you don't know and can't easily find other literature on this, might it be worth asking Doug Bates and / or José C. Pinheiro?

I'm working a relatively simple example to share with you, but I think I want to try a few things before I burden you with this.

Thanks for all you do to make it easier for people everywhere to get answers to difficult statistical questions.

Spencer Graves

On 5/16/23 3:38 PM, J C Nash wrote:

> There's a little example of how to get sv's from nls/minpack.lm etc. in the paper. I think it should
> be pretty easy to modify that to get something close to what you want.
>
> I'll be willing to take a look if you set up a "small reproducible example".
>
> Cheers, JN
>
> On 2023-05-16 16:33, Spencer Graves wrote:
>
> > Hi, John:
> >
> > On 5/16/23 2:45 PM, J C Nash wrote:
> >
> > > Take a look at just (re-)submitted review paper for R Journal that's at
> > > https://github.com/nashjc/RNonlinearLS/blob/main/RefactoringNLS/ComparingNLStools.pdf
> > >
> > > The "print()" generic for package nlsr puts out the singular values as an extra column.
> > > As we point out, these are lined up with parameters but are a property of the whole
> > > problem.

Wonderful.

> And (polite!) comments are welcome.

I'll look at that, but not right now.

How difficult might it be to get something similar from an alternative to optim?

As I mentioned, KFAS calls optim. I'd either have to reconfigure the KFAS maximum likelihood effort as a nonlinear least squares problem or modify one of your versions of optim to do the analysis you are doing with nlsr.

Thanks,
Spencer Graves

> Cheers, JN
>
> On 2023-05-16 13:00, Spencer Graves wrote:
>> Hi, John:
>>
>> What can you tell me about existing tools for diagnosing problems with nonlinear estimation with parameters that are redundant or nearly so?
>>
>> The singular value decomposition of the X matrix in linear regression will identify variables that are completely collinear or are sufficiently close that they cannot be well estimated independently of one another. The same could be done with a matrix of first partial derivatives of an objective function. I may have discussed this with you in the past, but I don't recall having gotten a satisfactory answer. I'm currently trying to use the KFAS package, which calls optim and does not have good diagnostics for this situation. I encountered this problem in 2016, and it's biting me again. I think I can do better with this now than I did in 2016, but it seems to me that this problem might be handled more easily if one of your alternatives to optim had better ways for handling this kind of thing.
>>
>> Comments?
>> Thanks,
>> Spencer Graves, PhD
>> Founder, EffectiveDefense.org
>> 4550 Warwick Blvd 508
>> Kansas City, MO 64111
>> m: 408-655-4567