Journal of Modern Applied Statistical Methods

Volume 4 | Issue 1 Article 31

5-1-2005

A Comparison of Nonlinear Regression Codes

Paul Fredrick Mondragon
United States Navy

Brian Borchers
New Mexico Tech

Journal of Modern Applied Statistical Methods
May, 2005, Vol. 4, No.1, 343-351

1538 – 9472/05/$95.00

A Comparison of Nonlinear Regression Codes

Paul Fredrick Mondragon
United States Navy
China Lake, California

Brian Borchers
Department of Mathematics
New Mexico Tech

Five readily available software packages were tested on nonlinear regression test problems from the NIST Statistical Reference Datasets. None of the packages was consistently able to obtain solutions accurate to at least three digits. However, two of the packages were somewhat more reliable than the others.

Key words: nonlinear regression, Levenberg – Marquardt, NIST StRD

Introduction

The goal of this study is to compare the nonlinear regression capabilities of several software packages using the nonlinear regression datasets available from the National Institute of Standards and Technology (NIST) Statistical Reference Datasets (National Institute of Standards and Technology [NIST], 2000).

The nonlinear regression problems were solved by the NIST using quadruple precision (128 bits) and two public domain programs with different algorithms and different implementations; the convergence criterion was residual sum of squares (RSS) and the tolerance was 1E-36. Certified values were obtained by rounding the final solutions to 11 significant digits. Each of the two public domain programs, using only double precision, could achieve 10 digits of accuracy for every problem. (McCullough, 1998).

Paul Mondragon is an Operations Research Analyst. Contact him at Paul.Mondragon@navy.mil. Brian Borchers is Professor of Mathematics. His research interests are in interior point methods for linear and semi-definite programming, with applications to combinatorial optimization problems. Contact him at borchers@nmt.edu.

The software packages considered in this study are:

1. MATLAB codes by Hans Bruun Nielsen (2002).
2. GaussFit (Jeffreys, Fitzpatrick, McArthur, & McCartney, 1998).
3. Gnuplot (Crawford, 1998).
4. Microsoft Excel (Mathews & Seymour, (1994).
5. Minpack (More, Garbow, & Hillstrom, 1980).

Hiebert (1981) compared 12 Fortran codes on 36 separate nonlinear least squares problems. Twenty-eight of the problems used by Hiebert are given by Dennis, Gay, and Welch (1977) with the other eight problems given by More, Garbow, and Hillstrom, (1978). In their paper, More et al. (1978) used Fortran subroutines to test 35 problems. These 35 problems were a mixture of systems of nonlinear equations, nonlinear least – squares, and unconstrained minimization. We are not aware of any other published studies in which codes were tested on the NIST nonlinear regression problems.

Methodology

Following McCullough (1998), accuracy is determined using the log relative error (LRE) formula,

$$\lambda_q = -\log_{10}\left[\frac{|q-c|}{|c|}\right]. \quad (1)$$

where q is the value of the parameter estimated by the code being tested and c is the certified value. In the event that $q = c$ exactly then λ_q is not formally defined, but we set it equal to the number of digits in c. It is also possible for an LRE to exceed the number of digits in c; for example, it is possible to calculate an LRE of 11.4 even though c contains only 11 digits. This is because double precision floating point arithmetic uses binary, not decimal arithmetic. In such a case, λ_q is set equal to the number of digits in c. Finally, any λ_q less than one is set to zero.

Robustness is an important characteristic for a software package. In terms of accuracy, there is concern with each specific problem as individuals. Robustness, however, is a measure of how the software packages performed on the problems as a set. In other words, there must be a sense of how reliable the software package is so there may be some level of confidence that it will solve a particular nonlinear regression problem other than those listed in the NIST StRD.

In this sense, robustness may very well be more important to the user than accuracy. Certainly the user would want parameter estimates to be accurate to some level, but accuracy to 11 digits is often not particularly useful in practical application. However, the user would want to be confident that the software package they are using will generate parameter estimates accurate to perhaps 3 or 4 digits on most any problem they attempt to solve. If, on the other hand, a software package is extremely accurate on some problems, but returns a solution which is not close to actual values on other problems, the user would want to use this software package with extreme caution.

The codes were not compared on the basis of CPU time, for the reason that all of these codes solve (or fail to solve) all of the NIST test problems within a few seconds. CPU time comparisons would certainly be of interest in the context of problems with many variables, or in problems for which the model and derivative computations are extremely time consuming.

A closer look at the various software packages chosen for this comparative study follows. Some of the packages are parts of a larger package, such as Microsoft Excel. In this case, the parts of the larger package which were used in the completion of this study are considered. Others in the set of packages used are designed exclusively for solving nonlinear least – squares problems.

HBN MATLAB Code

The first software package used in this study is the MATLAB code written by Hans Bruun Nielson (2002). Nielson's code can work with a user supplied analytical Jacobian or it can compute the Jacobian by finite differences. The Jacobian was calculated analytically for the purpose of this study.

GaussFit

GaussFit (Jeffreys et al., 1998) was designed for astrometric data reduction with data from the NASA Hubble Space Telescope. It was designed to be a flexible least squares package so that astrometric models could quickly and easily be written, tested and modified. In this study, version 3.53 of GaussFit was used.

A unique feature of GaussFit is that although it is a special purpose system designed for estimation problems, it includes a full-featured programming language which has all the power of traditional languages such as C, Pascal, and Fortran. This language possesses a complete set of looping and conditional statements as well as a modern nested statement structure. Variables and arrays may be freely created and used by the programmer. There is therefore no theoretical limit to the complexity of model that can be expressed in the GaussFit programming language.

One of the onerous tasks that faces the implementer of a least squares problem is the calculation of the partial derivatives with respect to the parameters and observations that are required in order to form the equations of

condition and the constraint equations. GaussFit solves this problem automatically using a built-in algebraic manipulator to calculate all of the required partial derivatives. Every expression that the user's model computes will carry all of the required derivative information along with it. No numerical approximations are used.

Gnuplot

Gnuplot (Crawford, 1998) is a command-driven interactive function plotting program capable of a variety of tasks. Included among these tasks are plotting both two- or three-dimensional functions in a variety of formats, computations in integer, floating point, and complex arithmetic, and support for a variety of operating systems.

The 'fit' command can fit a user-defined function to a set of data points (x,y) or (x,y,z), using an implementation of the nonlinear least-squares Marquardt – Levenberg algorithm. Any user-defined variable occurring in the function body may serve as a fit parameter, but the return type of the function must be real.

For this study, gnuplot version 3.7 patchlevel 3 was used. Initially, gnuplot displayed only approximately 6 digits in its solutions to the estimation of the parameters. The source code was modified to display 20 digits in its solutions. For the purposes of this study, FIT_LIMIT was set to 1.0e-15, with the default values for the other program parameters.

Microsoft Excel

Microsoft Excel is a multi-purpose software package. As only a small part of its capabilities were used during the process of this study, discussion of Excel is limited to its 'Solver' capabilities. The Excel Solver function is a self-contained function in that all of the data must be located somewhere on the spreadsheet. The Solver allows the user to find a solution to a function that contains up to 200 variables and up to 100 constraints on those variables. A Quasi-Newton search direction was used with automatic scaling and a tolerance of 1.0e-15. (Mathews & Seymour, 1994).

MINPACK

Minpack (More et al., 1980) is a library of Fortran codes for solving systems of nonlinear equations and nonlinear least squares problems. Minpack is freely distributed via the Netlib web site and other sources. The algorithms proceed either from an analytic specification of the Jacobian matrix or directly from the problem functions. The paths include facilities for systems of equations with a banded Jacobian matrix, for least squares problems with a large amount of data, and for checking the consistency of the Jacobian matrix with the functions.

For the problems involved in this study a program and a subroutine had to be written. The main program calls the lmder1 routine. The lmder1 routine calls two user written subroutines which compute function values and partial derivatives.

Results

The problems given in the NIST StRD dataset are provided with two separate initial starting positions for the estimated parameters. The first position, Start 1, is considered to be the more difficult because the initial values for the parameters are farther from the certified values than are the initial values given by Start 2. For this reason, one might expect that the solutions generated from Start 2 to be more accurate, or perhaps for the algorithm to take fewer iterations. It is interesting to note that in several cases the results from Start 2 are not more accurate based upon the minimum LRE recorded.

The critical parameter used in the comparison of these software packages is the LRE as calculated in (1). The number of estimated parameters for these problems range from two to nine. It was decided that it would be beneficial for the results table to be as concise as possible, yet remain useful. As a result, after running a particular package from both starting values, the LRE for each estimated parameter was calculated. The minimum LRE for the estimated parameters from each starting position was then entered into the results table.

Table 1. Minimum Log Relative Error of Estimated Parameters.

Problem	Start	Excel	Gnuplot	GaussFit	HBN	Minpack
Misra1a	1	4.8	5.8	10.0	11.0	7.7
	2	6.1	5.8	10.0	10.3	7.7
Chwirut2	1	4.2	4.9	7.4	10.6	2.4
	2	4.6	4.9	8.6	9.1	2.4
Chwirut1	1	4.0	4.2	8.0	10.3	7.5
	2	4.9	4.3	8.5	10.1	7.5
Lanczos3	1	0.0	3.9	0.0	4.9	3.3
	2	0.0	3.9	7.9	5.1	3.3
Gauss1	1	4.7	5.1	8.7	6.9	8.0
	2	4.6	5.1	8.6	6.9	3.3
Gauss2	1	4.5	4.9	0.0	6.8	7.8
	2	4.4	4.9	0.0	6.8	7.2
DanWood	1	4.6	5.1	NS	10.2	6.6
	2	4.7	5.1	NS	8.7	6.6
Misra1b	1	4.4	5.8	0.0	10.9	2.7
	2	6.4	5.8	9.7	11.0	2.5
Kirby2	1	1.0	4.8	7.4	10.3	6.2
	2	1.9	4.9	7.9	10.4	6.2

Problem	Start	Excel	Gnuplot	GaussFit	HBN	Minpack
Hahn1	1	0.0	4.0	0.0	9.5	NS
	2	0.0	4.0	0.0	9.7	NS
Nelson	1	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.0	1.4	0.0	0.0
MGH17	1	0.0	NS	NS	0.0	7.6
	2	1.4	3.7	NS	0.0	7.5
Lanczos1	1	0.0	10.0	0.0	4.9	4.3
	2	0.0	10.0	10.0	5.8	4.3
Lanczos2	1	0.0	5.4	0.0	5.7	3.5
	2	0.0	5.4	9.1	5.3	3.5
Gauss3	1	4.3	4.8	9.2	6.5	2.4
	2	4.1	5.0	9.1	6.5	2.4
Misra1c	1	0.0	5.9	0.0	10.8	7.6
	2	0.0	5.9	10.0	10.2	7.6
Misra1d	1	5.2	5.8	0.0	11.0	7.6
	2	4.4	5.9	8.9	11.0	7.6
Roszman1	1	3.5	4.1	8.7	4.0	0.0
	2	0.0	5.1	8.6	4.0	0.0
ENSO	1	0.0	1.6	3.7	6.5	0.0
	2	0.0	2.2	3.7	6.6	0.0

Problem	Start	Excel	Gnuplot	Gaussfit	HBN	Minpack
MGH09	1	0.0	3.6	0.0	5.0	6.3
	2	5.0	3.6	0.0	5.2	6.4
Thurber	1	1.7	3.2	0.0	7.8	0.0
	2	1.5	4.4	6.4	7.5	0.0
BoxBOD	1	0.0	4.5	NS	9.7	0.0
	2	5.6	3.8	NS	8.6	9.1
Rat42	1	5.3	4.2	8.0	10.3	7.1
	2	5.2	4.1	8.3	11.2	7.1
MGH10	1	0.0	NS	0.0	0.0	10.8
	2	0.0	4.4	0.0	0.0	11.0
Eckerle4	1	0.0	0.0	0.0	8.1	0.0
	2	5.1	4.8	8.3	7.2	1.2
Rat43	1	0.0	NS	NS	0.0	6.9
	2	3.2	2.6	NS	1.3	7.0
Bennett5	1	0.0	6.4	NS	3.7	0.0
	2	0.0	6.7	NS	3.7	1.5

Notes: NS – Software package was unable to generate any numerical solution. A score of 0.0 implies that the package returned a solution in which at least one parameter was accurate to less than one digit.

An entry of 0.0 in the results table is given if a software package generated estimates for the parameters but the minimum LRE was less than 1.0. For example if the minimum LRE was calculated to be 8.0e-1, rather than entering this, a 0.0 was entered. This practice was followed in an effort to be consistent with established practices (McCullough, 1998). If a software package did not generate a numerical estimate for the parameters, then an entry of 'NS' is entered into the results table.

Accuracy

As stated in the introduction, the accuracy of the solutions was evaluated in terms of the log relative error (LRE) using equation (1). Essentially the LRE gives the number of leading digits in the estimated parameter values that correspond to the leading digits of the certified values. Again, it should to be noted that the values given in the results table are the minimum LRE values for those problems. In other words, if a problem has five parameters to be estimated and four of the parameters are estimated accurately to seven digits, but the fifth is only accurate to one digit, it is reasonable to say that the problem was not accurately solved. On the other hand, if all five parameters were estimated to at least five digits, then one could feel confident that the package had indeed solved the problem.

Nielsen's MATLAB code had an average LRE score of 6.8 for the problems. For the problems this package was able to solve, the starting position did not seem to be of much importance. In fact, it is quite interesting that for several problems the LRE generated using the first set of initial values is larger than the LRE generated using the second set of initial values. This is interesting because the second set of initial values is closer to the certified values of the parameter estimates. Of the twenty-three problems that the parameters were estimated correctly to at least two digits, the average LRE was 7.96. This shows us that the accuracy of the estimated parameters was very high on those problems which this package effectively solved.

GaussFit had an average LRE score of 4.9. Unlike Nielsen's MATLAB code, GaussFit was very dependent upon the initial values given to the parameters. On eight of the problems GaussFit was unable to estimate all of the parameters to even one digit from the first starting position. From the second starting position GaussFit was able to estimate all of the parameters to over six digits correctly. This seemingly high dependence upon the starting values is a potential problem when using GaussFit for solving these nonlinear regression problems. There is no guarantee that one can find a starting value which is sufficiently close to the solution for GaussFit to effectively solve the problem.

Gnuplot has an average LRE score of 4.6. While this is actually lower than the average LRE score for GaussFit, gnuplot is not so heavily dependent upon the starting position in order to solve the problem. Rather, much like Nielsen's code, gnuplot seems quite capable of accurately estimating the parameter values to four digits whether the starting position is close or far from the certified values.

Microsoft Excel did not solve these problems well at all. The average LRE score for Excel is 2.32. Excel did perform reasonably well on the problems with a lower level of difficulty. For the eight problems with a lower level of difficulty the average LRE was 4.18. While these are probably reasonable results for these problems, we can see that for the problems with a moderate or high level of difficulty Excel did very poorly. Such results as this would cause one to have serious questions as to Excel being able to solve any particular least squares regression problem.

The Minpack library of Fortran codes also performed poorly on these particular problems. The average LRE for the twenty-six problems that Minpack did solve is 4.51. Minpack was significantly less accurate than the other packages on four of the problems, Misra1b, ENSO, Thurber, and Eckerle4. On the other hand, Minpack was considerably more accurate on the MGH10 problem. Minpack did not seem to be overly dependent upon starting position as in only two of the problems was there a significant difference in the minimum LRE for the different starting positions.

Table 2. Comparison of Robustness

Package	N	P(%)
Gnuplot	24	88.89%
Nielsen's MATLAB Code	23	85.19%
GaussFit	17	62.96%
Minpack	17	62.96%
Excel	15	55.56%

Robustness

Although the accuracy to which a particular software package is able to estimate the parameters is an important characteristic of the package, the ability of the package to solve a variety of nonlinear regression problems to an acceptable level of accuracy is perhaps more important to the user. Most users would like to have confidence that the particular software package in use is likely to estimate those parameters to an acceptable level of accuracy.

What is an acceptable level of accuracy? Such a question as this might elicit a variety of responses simply depending upon the nature of the study, the data, the relative size of the parameters, and many other variables which may need to be considered. For the purposes of this study we will consider an acceptable level of accuracy to be three digits. In Table 2, the various software packages are compared by the number (and percentage) of the problems which they were able to estimate the parameters accurately to at least three digits from either starting position.

Here, N is the number of problems which the package accurately estimated the parameters to at least three digits. P is the percentage of the problems which the package accurately estimated the parameters to at least three digits.

It can easily be seen here that as far as the robustness of the packages is concerned there are two distinct divisions. Nielsen's MATLAB code, and Gnuplot were both able to attain the 3 digit level of accuracy for over 80% of the problems. GaussFit, Excel, and Minpack, on the other hand were able to attain that level of accuracy on less than 65% of the problems.

Conclusion

The robustness of the codes tested in this study is surprisingly poor. In many cases, the results were quite accurate from one starting point, and completely incorrect from another starting point. In some cases the codes failed with an error message indicating that no correct solution had been obtained, while in other cases an incorrect solution was returned without warning.

Although some problems seemed to be easy for all of the codes from all of the starting points, there were other problems for which some codes easily solved the problem while other codes failed. In general, when reasonably accurate solutions were obtained, the solutions were typically accurate to five digits or better.

It is suggested that users of these and other packages for nonlinear regression would be well advised to carefully check the results that they obtain. Some obvious strategies for checking the solution include running a code from several different starting points and solving the problem with more than one package.

References

Crawford, D. (1998). Gnuplot Manual. Retrieved March 4, 2004, from www.ucc.ie/gnuplot/gnuplot.html.

Dennis, J. E., Gay, D. M., & Welch, R. E. (1997). An adaptive nonlinear least – squares algorithm, NBER working paper 196, M.I.T./C.C.R.E.M.S., Cambridge, Mass.

Hiebert, K. L. (1981). An Evaluation of Mathematical Software That Solves Nonlinear Least Squares Problems, *ACM Transactions on Mathematical Software*, *7*(1), 1-16.

Jefferys, W. H., Fitzpatrick, M. J., McArthur, B. E., & McCartney, J. E. (1998). *GaussFit: A system for least squares and robust estimation users manual*. Austin: University of Texas.

Kitchen, A. M., Drachenberg, R., & Symanzik, J. (2003). Assessing the reliability of web-based statistical software, *Computational Statistics*, *18*(1), 107-122.

Mathews, M., & Seymour S. (1994). *Excel for Windows: The Complete Reference*. (2nd ed.). NY:McGraw-Hill Inc.

McCullough, B. D. (1998). *Assessing reliability of statistical software: Part I: The American Statistician*, *52*(4), 358-366.

More, J. J., Garbow, B. S., & Hillstrom, K. E. (1978). Testing unconstrained optimization software. Rep. TM-324, Applied Math Division, Argonne National Lab., Argonne, IL.,

More, J. J., Garbow, B. S., & Hillstrom, K. E. (1980). User guide for MINPACK-1. Report ANL-80-74. Argonne, IL: Argonne National Lab.

National Institute of Standards and Technology. (2000). Statistical Reference Datasets (StRD). Retrieved March 4, 2004 http://www.itl.nist.gov/div898/strd/.

Nielsen, H. B. (2002). Nonlinear Least Squares Problems. Retrieved March 4, 2004 from www.imm.dtu.dk/~hbn/Software/#LSQ.