MODPAC: A modular package of programs for fitting model parameters to data and plotting fitted curves

ADAM V. REED

The Rockefeller University, New York, New York 10021

A modular package of computer programs is described. The package is designed for parameter fitting in psychology, and includes a program for plotting the fitted curves.

In comparing the results of psychological experiments to theoretical models of the underlying mental processes, it is often necessary to determine what values of the model's free parameters optimize its fit to the data. Similar problems also arise in the physical sciences, and computer programs developed in physical science contexts are often used by psychologists. These programs, however, are sometimes less than ideally suited for parameter fitting in psychology. In most cases the parameter-fitting programs developed for the physical sciences use Newton's method to accelerate the hill-climbing process. While Newton's method drastically reduces the computational effort necessary to find optimum parameter values, it is applicable only to smooth functions. Some psychological models lead to functions which are not in that category. The modular package of programs presented here was designed specifically for parameter fitting in psychological contexts. The package also includes a program for plotting the fitted curves, producing a publication-ready figure. The modular design permits the user to determine which error measure he wishes to use and to compare parameter fits using several error measures with the same model and data. It also readily permits goodness-of-fit comparisons of several models with the same set of data, comparisons of several sets of data to a given model, etc., with a minimum number of monitor commands.

The present version of the parameter-fitting package is written in DEC PDP-15 FORTRAN IV for implementation on-line on the DEC PDP-15 computer under the DOS monitor. This implementation is close to the standard versions of FORTRAN IV, and users of other machines can readily adapt the program to their systems. The package is designed for users who can write their own model prediction subroutines and modify other supplied programs according to their own requirements. It is available from the Digital Equipment Computer Users Society, 146 Main Street, Maynard, Massachusetts 01754; order DECUS No. 15-92.

This work was supported by Alcohol, Drug Abuse, and Mental Health Administration National Research Service Award IF32MH05028-01 from the National Institute of Mental Health.

PROGRAM PACKAGE DESCRIPTION

The program package includes four types of program modules: main hill-climbing module, error computing subroutine modules (ERCOM), dependent variable predicting (model implementing) function modules (DEVAR), usually user generated, and plotting modules. Each type of module will be considered in turn.

Main Hill-Climbing Module

The function of the main hill-climbing module is to find which values of the model parameters best fit the data, that is, to find those values for which the squared error measure (SERSQ) is minimal. The main hill-climbing module communicates with data file PARAM.SRC and with the error computing module (subroutine ERCOM). The data file PARAM.SRC contains, in sequential records or groups of records, the number of the parameters to be adjusted (N), the starting values of the N parameters [A(I)], the lower limits [ALWLM(I)] on parameter values, and the corresponding upper limits [AUPLM(I)]. The initial call to ERCOM has arguments (A, $\emptyset$, N) passing the number of climb parameters to be used in computing subsequent goodness-of-fit measures. Subsequent calls to ERCOM have arguments (A, SERSQ, $\emptyset$), passing the current values of the parameters to the subroutine and getting the error measure back. The final call has arguments (ALAST, VLAST, -1), and is used to compute a goodness-of-fit measure for the fitted approximation and to type the final report to the user.

The modular structure of the program package permits the use of as many different main hill-climbing modules as one wishes to write, possibly using different methods of finding parameter optima. The only required commonality is in communication with PARAM.SRC, ERCOM, and, if plotting is to be done, with output file FOUND.SRC. The attached version of the main hill-climbing module alternates partial grid search along parameter axes with linear extrapolation after two loops of five hill-climbing iterations each. During each hill-climbing iteration, the current values and grid step sizes for all the parameters are adjusted

in turn. For each parameter adjustment, the error is sampled at five values of the parameter with the other parameters held constant. The optimum value of the parameter is taken as its new current value. The grid size is divided by three if one of the middle points is optimal; if one of the extreme points is optimal, it is multiplied by three. The subsequent linear extrapolation follows the line from parameter coordinates after the first loop through parameter coordinates after the second. Thus, this version of the main hill-climbing module will find one of the logical optima, relative to movement along parameter axes, within the segment of parameter space delimited by ALWLM and AUPLM.

This hill-climbing module works on, at most, 20 parameters. It may be expanded to take care of a larger number by changing the dimensions of the arrays in the first DIMENSION statement. Similarly, the number of iterations per loop may be changed by editing the LOOP = statement. There are two other limitations of this version. With some functions, it is possible for a point in parameter space to be optimal with respect to movement parallel to parameter axes, without being optimal with respect to movement in some other direction. Such functions are seldom encountered in psychological work, and may be dealt with by writing an alternative hill-climbing module with complete instead of partial grid search. Second, one may have more than a single local optimum in the segment of parameter space under consideration. The present program, like all hill-climbers, will find only one. If the presence of additional optima is suspected, a variety of starting points may be tried.

Error Computing Module

The function of the error computing module is to compare the values of the dependent variable predicted by the model under the assumption that the model parameters have the values transmitted by the main hill-climbing module with the given empirical values of the dependent variable, as measured experimentally. The error computing module, ERCOM, obtains the experimental data from data file GIVEN.SRC, and the predicted values of the dependent variable from the dependent variable computing module function DEVAR. ERCOM also labels the program's output, and types out its final report to the user. The specifications of the arguments of calls to the subroutine ERCOM from the main hill-climbing module are given in the description of the main module above.

ERCOM modules may be designed for scalar or vector functions of any number of variables, and for any computable error measure. The three versions supplied with the package are designed for continuous (floating point) functions of one continuous (floating point) and one discrete (integer) variable. They may be easily edited, however, to be used with functions of any number of

floating point or discrete variables. Each of the three attached versions of ERCOM computes a different error measure. ERLIN, the first version, computes the total squared linear error, which is the usual error measure in most psychological work. ERLOG does the same with squared logarithmic error, and may be used simultaneously on several sets of data, measured in different regions of the dependent variable, without the need for rescaling. ERLIN also computes r^2 according to the formula:

$$r^2 = 1 - \frac{\sum_{j=1}^m (\hat{x}_j - x_j)^2 / (m - k)}{\sum_{j=1}^m (x_j - \bar{x})^2 / (m - k)},$$

where x_j are the M empirical measurements of the dependent variable, and $\hat{x}_j$ are the corresponding model predictions using the k fitted free parameters. A similar logarithmic r^2 is computed by ERLOG. The third version, ERMMX, passes to CLIMB the square of the largest single discrepancy between a data point and its corresponding model prediction. Other ERCOM subroutines may be devised for other error measures, including, for example, weighted measures for use when the different data points result from different numbers of data measurements.

On the first call to ERCOM, each version of the subroutine reads in from GIVEN.SRC the number M of empirical points, the corresponding M values of the floating point independent variable $X(J)$, the M empirical values $E(J)$ of the dependent variable, and the corresponding values of the integer variable $NEQ(J)$. When working with functions of one independent variable, the other independent variable is treated as a dummy variable. On subsequent calls, ERCOM receives the current values of the to-be-fitted parameter, and, for each data point, sends them along with the values of the independent variables to the function subroutine DEVAR, which returns the predicted value of the dependent variable. This value is compared to the empirical value for each point and the error measure sent back to CLIMB. On the final call, a report—possibly including r^2 —is computed and typed out.

Dependent Variable Function Module

This module will be written by the user to generate the predicted values of the dependent variable, as a function of the independent variables NEQ and X , and of the parameters in array A . One possible DEVAR function, incorporating the model of Hypothesis 12 of Reed (1976), is supplied with the package as an example. If the function is to be used with the ERCOM subroutines supplied in the package, its first two lines should be

FUNCTION DEVAR (NEQ,X,A)
 DIMENSION A(20)

and it must incorporate a DEVAR = statement and a RETURN statement.

Plotting Modules

The data file GIVEN.SRC may be used to plot the data points using the plotting program PTSPL. Several "point" shapes are used for points associated with different values of NEQ. A second plotting program, CURPL, also uses GIVEN.SRC to plot the data points and, in addition, plots the fitted curves for each value of the integer independent variable present in the data. The latter program uses the same function DEVAR as used by ERCOM. It obtains the parameter array from file FOUND.SRC, created by CLIMB. These programs will work without modification on installations equipped with J. Wyatt's plotter and display output routines, DECUS No. 9-13. Both programs plot the continuous independent variable on the abscissa and the dependent variable on the ordinate. The size of the plot and the labels of the coordinate axes are obtained from the user's teletypewriter during execution.

Data Files

The files PARAM.SRC, GIVEN.SRC, and FOUND.SRC from the example session described in the write-up are included in the package as examples for the user. In normal usage the GIVEN file would be stored under a mnemonic associated with the experiment in which the data it contained were obtained (e.g., GVL76.SRC) and transferred to GIVEN.SRC using PIP before use.

USING THE MODULAR PACKAGE

On A PDP-15 Computer

The BIN files of the programs one wishes to use, as well as the necessary data files (GIVEN.SRC, PARAM.SRC for CLIMB, FOUND.SRC if CURPL is to be used without using CLIMB first) must be placed on the system device. For parameter fitting, GLOAD should be followed by CLIMB, the name of the version of ERCOM one wishes to use, and the name of the ver-

Figure 1. An example of plot produced by CURPL.

sion of DEVAR one wishes to try, for example, CLIMB, ERMUX, DVH12, for minimax fitting of Hypothesis 12. To plot the curves resulting from a fitting run, one invokes GLOAD again, followed by CURPL and the name of the same version of DEVAR as in the fitting run immediately preceding, for example, CURPL, DVH12.

The protocol of a sample fitting session, using programs supplied with the package, is provided in DECUS write-up 15-92. A sample CURPL plot is shown in Figure 1.

Implementation on Other Systems

While a considerable amount of adaptation may be required to adapt the plotting programs to other computer-plotter systems, the rest of the package should be subject to implementation on any file-oriented system with only minor changes. In either case, the adaptation should be within the capability of any user sophisticated enough to be able to use the package properly.

REFERENCE

- REED, A. V. List length and the time course of recognition in immediate memory. *Memory & Cognition*, 1976, 4, 16-30.

(Received for publication April 13, 1976;
 revision accepted May 18, 1976.)