UCRL 10153

University of California

Ernest O. Lawrence Radiation Laboratory

A GENERAL PROGRAM FOR STATISTICAL ANALYSIS USING THE MAXIMUM-LIKELIHOOD METHOD. "MALIK" PROGRAM

Berkeley, California

DISCLAIMER

UCRL-10153
UC-32 Mathematics and Computers
TID-4500 (17th Ed.)

UNIVERSITY OF CALIFORNIA
Lawrence Radiation Laboratory
Berkeley, California

Contract No. W-7405-eng-48

A GENERAL PROGRAM FOR STATISTICAL ANALYSIS USING THE MAXIMUM-LIKELIHOOD METHOD. "MALIK" PROGRAM

Fernand Grard

March 30, 1962

A GENERAL PROGRAM FOR STATISTICAL ANALYSIS USING THE MAXIMUM-LIKELIHOOD METHOD. "MALIK" PROGRAM

Fernand Grard

Lawrence Radiation Laboratory
University of California
Berkeley, California

March 30, 1962

ABSTRACT

A program for analysis of experimental data by the maximum-likelihood method is described. The solution, along with the corresponding error matrix, is determined by a stepping procedure. The program can be adapted easily to any particular problem, by coding two simple subroutines appropriate to that problem.

A GENERAL PROGRAM FOR STATISTICAL ANALYSIS USING THE MAXIMUM-LIKELIHOOD METHOD. "MALIK" PROGRAM

Fernand Grard †

Lawrence Radiation Laboratory
University of California
Berkeley, California

March 30, 1962

1. Introduction

This note describes a program which has been written for the analysis of experimental data by use of the maximum-likelihood method. The procedure used to find the solution consists of determining the maximum of the appropriate likelihood function and the error matrix by a method of steps. For any particular problem, the user of this program has to write two subroutines:

(a) a subroutine (INPUT) to read and store in the computer all the experimental information,

(b) a subroutine (FUNCT) in which the logarithm of the likelihood function is calculated from the experimental quantities and the parameters of the probability function to be fitted.

Although the program needs more time on the computer than when the solution is found by solving the system of the first derivatives of the likelihood function set equal to zero, [1] it presents the great advantage of being easily adapted to any particular problem without requiring that all the first and second derivatives of the likelihood function be worked out. The program called MALIK has been written for the 709 and 7090 IBM computers of the Lawrence Radiation Laboratory, Berkeley. It has to be run with the Fortran MONITOR system. A copy is available from Robert Harvey, Alvarez group, Berkeley. (Some other Fortran programs for maximizing functions may also be obtained from Robert Harvey).

2. Description of the Main Program

The main program is supposed to be used without any alteration for all cases. Assume that the probability function is represented by

$$f = f(a_1, a_2, \cdots a_n, X) \quad , \tag{1}$$

where $a_1, a_2, \cdots a_n$ are the n parameters to be determined. X is a symbol representing all the measured quantities.

---

† On leave from the Institut Interuniversitaire des Sciences Nucléaires, Brussels, Belgium.

[1] An example of this method is given in UCRL-10100.

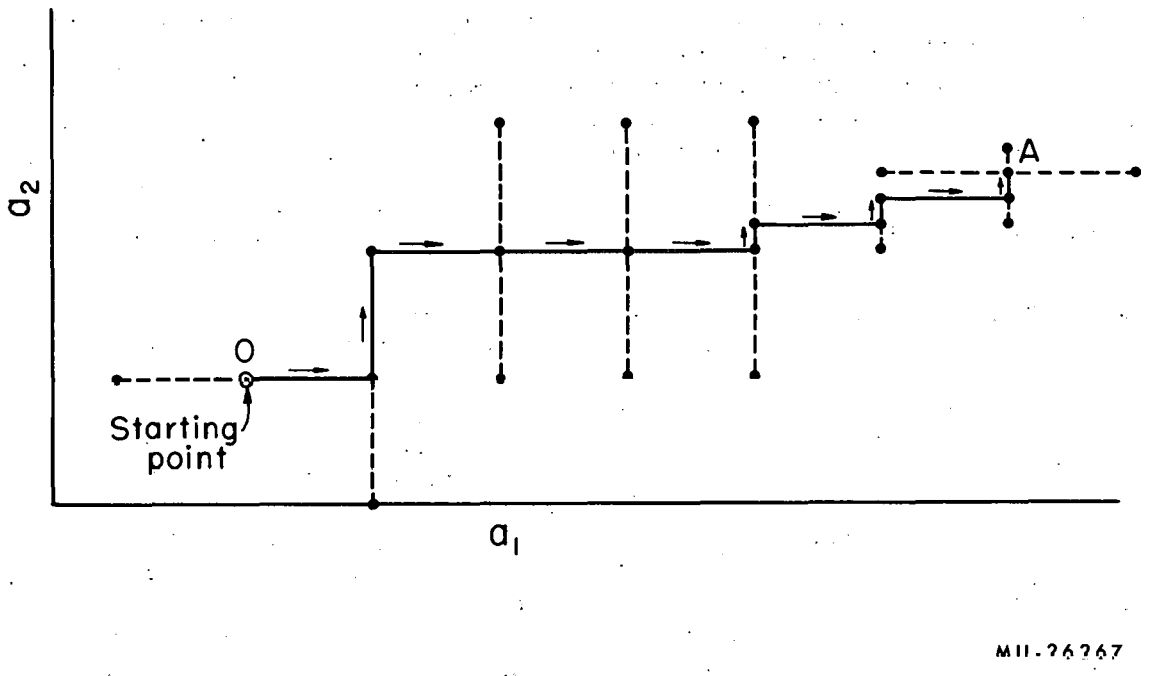

Fig. 1. Illustration of the described method to determine the maximum of the likelihood function. The factor α has been taken equal to 0.2. The point A corresponds to the largest value of the likelihood function one could reach with the assigned step sizes during the first iteration. The points joined by dotted lines represent unsuccessful attempts to start the second iteration, only the step assigned to parameter a_1 will be reduced by the factor α.

The likelihood function can be written

$$L = \prod_{i=1}^{N} \left[f(a_1, a_2, \cdots a_n, X_i) / \int_{(X)} f \, dX \right], \qquad (2)$$

the integration extending over the whole range of physical quantities where the measurements have been made. N represents the total number of events. The function F to be maximized is the logarithm of the likelihood function: $F = \ell nL$. The procedure starts from initial values of the parameters $a_1, a_2, \cdots a_n$. For each of them, a step of appropriate size is assigned by the user. The method used to reach the maximum is the following: the first parameter is increased or decreased by a quantity corresponding to its assigned step in order to find a larger value of the function F. If a larger value of F is successfully found, the corresponding new value of this first parameter is used and the same operation is repeated for the next parameter, and so on up to the last one. The cycle is repeated as long as the function F increases. When no improvement can be made with the step sizes being used, the steps are all reduced simultaneously by a common factor α, and the next iteration is started. The steps are reduced further and further until a satisfactory solution is found for the parameters $a_1, a_2 \cdots a_n$. This factor α is fed in the computer as initial information.

In the present version of the program, the above principle has been altered in the following way: If a parameter does not change after three consecutive attempts the corresponding step size is then reduced. This operation can be done only once for a given parameter during a given iteration, and can be extended to n-1 parameters. At the beginning of the next iteration, all the steps except those which have already been altered are reduced by the factor α. Figure 1 shows, for a two-parameter problem, how the maximum is searched for by use of this procedure.

Convergence Check

The convergence check has been chosen by relying on the fact that the likelihood function approaches a multi-Gaussian distribution. At the end of each iteration, the largest value obtained for the log of the likelihood function is compared with that resulting from the previous iteration, i.e., before the last reduction of the step has taken place. The iteration procedure is stopped when the difference between these two quantities is less than 0.001. It can therefore be said that the two corresponding solutions differ by less than a 1/20 of a standard deviation. Taking into account the further reduction of the steps (a factor of 0.1 is generally taken), one sees that the solution finally obtained can be very close to the maximum.

Error matrix

The error matrix is computed by inversion of the matrix built with all the second derivatives of the function F at its maximum. The second derivatives are calculated by differences.

In order to find an adequate size for the steps to be used, the width of the distribution in the direction of each of the parameters is first determined

as being the positive and negative quantities one has to add to each of the fitted parameters a to lower the likelihood function by $e^{-1/2}$ of its maximum value. With the assumption that the function L is a multi-Gaussian distribution, and if there were no correlation between the parameters, these quantities would be standard errors. They are printed among the final results, since they give useful information about the skewness of the likelihood function. The steps used to compute the second derivatives have been taken equal to a certain fraction β of the "standard errors" defining the width of the likelihood function. Although for a perfect multi-Gaussian distribution the results obtained for the second derivatives would be independent of the size of the step chosen to compute them, for practical cases it is advisable to take these steps as small as possible within the accuracy of the computer. In general, a value of $\beta = 0.2$ will still give significant differences. For a nonsymmetric likelihood function, the resulting second differences could differ from the second derivatives by nonnegligible third-derivative terms. Taking into account that these terms are odd with respect to the increments, the second differences are first computed for positive and for negative increments and the error matrix is then determined by inversion of the second derivatives matrix built with the mean values of these second differences. The factor β is fed into the computer as initial information.

REMARK

During the calculation of the second differences, values of the logarithm of the likelihood function are computed by changing two parameters simultaneously from the minimum, whereas in the search for the solution, the parameters are varied one at a time. It could therefore happen that a value of the likelihood function could be found higher than that corresponding to what has been called the satisfactory solution, if the iterations had stopped at a saddle point. The iteration procedure is then allowed to restart with the initial step size and continue until a new solution is found.

3. Operation of the program

The main program contains "Common" and "Dimension" statements that must be repeated in the two subroutines INPUT and FUNCT used in connection with it:

```
COMMON      PAR

DIMENSION   PAR(10)
```

The variables PAR are the parameters $a_1, \cdots a_n$ of Eq. (1) to be determined. Their number must not be greater than 10.

The program operates in the following way.

First, data cards with initial information are read into the computer. They include the following.

a. One card with the number of parameters to be fitted. Format: (20X, I10).

b. One card for each of the parameters with its initial value and the corresponding assigned step. Format: (10X, 2F10.5).

c. One card with the factor α used to reduce the steps at the end of each iteration, the total number of iterations allowed, and the factor β. Format: (20X, F10.5, I10, F10.5).

The main program then calls the subroutine INPUT with which the experimental data will be read and stored according to formats decided by the user. Thereafter the maximum of the function F and the error matrix are determined, calling the subroutine FUNCT whenever the value of the function is needed for a new set of the parameters.

When the problem is terminated, the program starts again ready to solve another problem. A new set of cards has to be prepared as indicated in a, b, and c above.

If no other problem has to be solved, a blank card set in place of the one in a causes the program to call EXIT.

4. Subroutines INPUT and FUNCT

The experimental data read by the subroutine INPUT are used exclusively by the subroutine FUNCT. These two subroutines must therefore have appropriate common statements. In the subroutine FUNCT, K represents the number of parameters and F is the logarithm of the likelihood function. These subroutines are to be coded as follows.

(a)
```
SUBROUTINE INPUT
DIMENSION PAR (10)
DIMENSION X(-)
COMMON PAR
COMMON X
[Read and store the measured quantities X(-)]
RETURN
END
```

(b)
```
SUBROUTINE FUNCT (F, K)
DIMENSION PAR (10)
DIMENSION X(-)
COMMON PAR
COMMON X
[Compute F, the logarithm of the likelihood function, using the measured
quantities X(-) and the parameters PAR(-)]
RETURN
END
```

5. OUTPUT

The following information is printed:

1) The step sizes at the beginning of each iteration

STEPS = ...

2) During a given iteration, the value of the logarithm of the likelihood function at the end of each cycle, i.e. when all the parameters have been considered consecutively, along with the last values of the parameters:

L.F. - ..., followed by

$a_1, a_2, a_3, \cdots a_n$.

3) The parameters, when a satisfactory solution has been reached:

THE LAST SOLUTION IS SATISFACTORY,

$a_1, a_2, \cdots a_n$.

4) The positive and negative increments defining the width of the likelihood function, for each parameter:

WIDTH OF LIK. FNCT. =

5) The two second-derivative matrices built with the second differences obtained from positive increments and from negative increments.

6) The error matrix as the inverse of the mean of the two previous matrices.

ACKNOWLEDGMENTS

The author is indebted to Dr. Ronn R. Ross and Dr. Frank Solmitz for helpful discussions. He also wishes to thank Prof. Arthur Rosenfeld and Prof. Lynn Stevenson for their interest in this work.

This work was done under the auspices of the U. S. Atomic Energy Commission.