A Least Squares Surface Fitting Program

By J. H. Cadwell

A program designed to fit a bivariate polynomial to a set of z values specified at points of a rectangular grid in the (x, y) plane is described. It uses the method of orthogonal polynomials, providing a detailed guide as to which polynomial terms need be included.

Introduction

This note describes a Mercury Autocode program prepared at the Royal Aircraft Establishment. It fits a function $f(x, y)$ to values of a variable z specified at points of a rectangular grid in the (x, y) plane. The grid spacing need not be uniform in either direction, and unequal weights can be allotted to rows, to columns, or to both. While the set of values will usually be complete, an iterative facility deals with the case where some are unspecified.

The function used is a bivariate polynomial, and the method discussed by Forsythe (1957) is used to determine its coefficients. Clenshaw (1960) has recently discussed a modification of the method using Chebyshev polynomials; this program uses the basic orthogonal forms. In the case of equally spaced points the orthogonal method has been widely used for curve fitting (e.g. Fisher and Yates, 1957). DeLury (1950) has also applied it to fitting a surface. The method offers distinct advantages in curve fitting, and it is the purpose of this note to show that, when fitting a surface, these advantages are still more pronounced.

It is well known that the normal equations that arise in the usual form of polynomial fitting are very ill-conditioned; the orthogonal method, however, leads to a set of simple equations. The other main advantage lies in the step-by-step nature of the process, and the fact that it provides a measure of the improvement resulting from each new term included. Kendall (1951) gives an algorithm for dropping a term, or adding a new one, in the standard method. However, it is too involved to be of much help when many possible terms can be included.

Let the maximum degree of x that occurs in the polynomial be t. We have to select integers $j(i)$ for i from 0 to t so that the coefficient of x^i is of degree $j(i)$ in y. It is natural to make the restriction

$$j(i) \geqslant j(i+1) \qquad (1)$$

or perhaps the more stringent

$$j + i = \text{constant}.$$

The program first fits all terms up to degree 8 in x and y separately, and finds the reduction in the sum of squares of residuals due to each of the 81 possible coefficients. The sum of these 81 quantities plus the sum of squares of residuals equals the sum of squares of the z values. They are described as *components* of the sum of squares, and they serve to indicate which terms are necessary, and which make only a trivial contribution to accuracy of fit. This information, together with a knowledge of the desired accuracy in fitted values, enables the best choice of polynomial terms obeying restriction (1) to be made. The program is then run again with the initial data tape suitably modified. It now calculates the chosen polynomial coefficients, and the residual values, i.e. the discrepancies between the original data and values predicted by the fitted function.

An Illustrative Example

As outlined above, the program has two quite distinct stages. Let there be m values of x and n values of y. The first stage fits all possible terms up to degree u in x and v in y, where

$$u = \min(8, m-1)$$
$$v = \min(8, n-1).$$

It then prints the sum of squares of residuals, and a table giving the reduction in the sum of squares of the original values for each of the $(u+1)(v+1)$ possible orthogonal components.

The following example illustrates the method, Table 1 being fitted by the first stage up to degree 4 in x and 8 in y. It is usually advisable to determine a polynomial in $(x - \bar{x})$ and $(y - \bar{y})$ rather than in x and y, as this greatly reduces the loss of significant figures in evaluating the fitted polynomial. Thus, in the present insta nce

Table 1

Function Values

y \ x	0	20	40	60	80
4·0	252·8	255·6	257·7	259·4	260·4
4·1	255·8	259·2	261·7	263·7	265·2
4·2	256·1	260·2	263·2	265·7	267·8
4·3	252·6	257·6	261·5	264·6	267·3
4·4	246·8	252·7	256·9	260·6	263·6
4·5	239·6	246·2	250·6	254·3	257·4
4·6	231·8	238·6	243·5	247·0	250·1
4·7	223·3	230·6	235·6	238·9	241·8
4·8	214·6	222·1	227·0	230·5	233·1
4·9	205·6	213·4	218·3	221·6	223·9
5·0	196·3	204·3	209·3	212·4	214·6

Table 2

Components of Sum of Squares

POWER OF y	POWER OF x				
	0	1	2	3	4
0	3·248, 6	1·588, 3	5·120, 1	1·657, 0	2·630, −2
1	1·669, 4	7·400, 1	8·365, 0	4·561, −1	1·875, −3
2	2·120, 3	2·016, 1	1·325, −1	1·462, −2	3·264, −5
3	1·855, 2	1·492, −2	6·927, −2	1·576, −4	1·228, −3
4	2·054, 0	1·031, 0	2·497, −6	6·853, −4	5·115, −4
5	1·708, 0	2·266, −1	2·637, −3	6·160, −3	1·669, −2
6	7·209, −1	1·030, −4	6·623, −4	8·386, −4	1·353, −2
7	4·507, −2	8·086, −2	9·183, −5	1·126, −3	7·823, −3
8	2·465, −3	2·143, −2	5·067, −3	7·845, −4	2·036, −2

Sum of squares of residuals, 4·090, −2

a polynomial in x, y would require coefficients correct to 10 significant figures to avoid rounding errors greater than 0·1 in its evaluation. With the transformed variables the same accuracy is attained with coefficients accurate to 6 significant figures. The program gives warning when rounding errors become serious.

Table 2 illustrates the results of stage 1. The floating-point specification of numbers is such that a, b implies $a.10^b$. It will be seen that fitting up to degree 4 in x and 8 in y gives a sum of squares of residuals of 0·041. We can also deduce the larger sum of squares of residuals that will arise if fitting is carried out to lower degrees in x and y.

Thus, if we consider fitting only powers lying above the dotted lines, the sum of squares of residuals becomes 0·041 plus each of the terms omitted, giving the quantity 0·381. A polynomial

$$\sum_{i=0}^{t} \sum_{j=0}^{j(i)} a_{ij}(x - \bar{x})^i(y - \bar{y})^j$$

where we use the values

$$t = 3,\quad j(0) = 6,\quad j(1) = 5,\quad j(2) = 2,\quad j(3) = 1,$$

will correspond to the dotted lines. After fitting in terms of the corresponding orthogonal components, these are multiplied out and coefficients collected up. The result is obtained from stage 2 of the program. The data tape for this second stage will be as for stage 1, but with the numerical values of t and the j's added. The results are given in Table 3.

The choice of the dotted lines is dictated by the following considerations.

(1) $j(0) \geqslant j(1) \geqslant \ldots \geqslant j(t)$.

(2) The sum of squares of residuals should be as small as possible consistent with a given total number of coefficients.

When dealing with data subject to random effects the statistician would use a further criterion. This depends on the mean square residual defined as the sum of squares of residuals divided by $mn - (u + 1)(v + 1)$. Individual terms in Table 2, smaller than two or three times this quantity, would be disregarded, as far as consideration (1) above allowed.

It will be seen that the maximum residual for each of the eleven y values is also tabled. An alternative print-out gives all residuals, should these be required.

Residuals are evaluated directly from the final polynomial and, should they be affected by rounding errors, this fact will appear from the incompatibility of the sum of squares of residuals and the check value. This latter quantity is found independently, and is not affected by the loss of significant figures that may occur when subtracting nearly equal quantities in polynomial evaluation.

Provided the listed residuals are acceptably small, a discrepancy between the two quantities is unimportant. Otherwise a change of origins, or a reduction in the degrees used, can be tried. It may seem paradoxical to expect a better fit by using lower-degree polynomials. However, rounding error effects usually increase rapidly with degree, while beyond a certain point residuals decrease rather slowly. Thus the choice of degrees larger than are justified by the data may well increase residuals when working in single-length arithmetic.

In some applications it is useful to be able to allot different weights to either the x columns, the y rows, or to both, and the program allows this to be done. When either x or y values are in arithmetic progression, only the first value(s) and common difference(s) are punched.

Table 3

Polynomial Coefficients

POWER OF x	POWER OF y	COEFFICIENT	
0	0	2·50746554,	2
	1	−6·96366345,	1
	2	−5·05008445,	1
	3	1·14543928,	2
	4	−1·55554895,	2
	5	−1·16986970,	2
	6	3·28594729,	2
1	0	2·02546036,	−1
	1	6·77372002,	−2
	2	−4·40136945,	−1
	3	5·01602247,	−1
	4	7·91083973,	−1
	5	−1·50640930,	0
2	0	−1·35851656,	−3
	1	−1·84253163,	−3
	2	−8·30420598,	−4
3	0	1·27840913,	−5
	1	2·12121172,	−5

Maximum Residuals

y	RESIDUAL		x
1	1·017,	−1	4
2	−1·256,	−1	1
3	2·216,	−1	1
4	−8·689,	−2	2
5	−1·766,	−1	3
6	1·500,	−1	2
7	1·240,	−1	3
8	1·497,	−1	3
9	−1·136,	−1	1
10	8·991,	−2	4
11	−7·729,	−2	3

Sum of squares of residuals	3·813,	−1
Check value	3·813,	−1

Missing Values

Should there be some values missing from the rectangular array we can proceed as follows. For stage 1, appropriate guessed values are substituted, and the results are used to provide the required values of t and of the j's for stage 2.

This now becomes an iterative process. After fitting to the data, the guessed values are replaced by the values predicted from the fitted surface. This process continues until the differences between consecutive sets of predicted values are less than a prescribed amount. The final fitted coefficients, together with residuals and the estimated missing values, are printed. This method, sometimes called the "missing plot" technique, provides the correct least squares solution of the original problem.

In this iterative process the problem of convergence arises, and two points, besides the obvious requirement of good starting values, have to be watched. The degrees used should not be larger than are justified by the data. In addition, the accuracy sought in predicted values should not be too great. The square root of the mean square residual defined earlier is a suitable value. The program prints a unit once in each iterative cycle, and so there is available a record of the number of iterations required, enabling the process to be watched while the program is running.

Working Times

The basic curve-fitting process uses only the high-speed store, taking about 8 seconds to fit 100 points up to degree 8. On this basis the surface-fitting procedure takes about $0{\cdot}08n(m + 9)$ seconds. To this must be added the time taken in polynomial evaluation and in reading data and punching results. In view of the number of possible variants these times are difficult to predict accurately. Table 4 illustrates stage 2 times in three typical instances. They are for the case where maximum residuals are printed.

Table 4

SIZE OF ARRAY	WORKING TIME (SECONDS)	READ AND PUNCH TIME (SECONDS)
15 × 10	34	74
30 × 30	116	180
100 × 48	470	530

Times for stage 1 are rather smaller; where missing values arise, the working time has to be multiplied by the number of iterations required.

The program will deal with up to 50 x values and 48 y values in the general case. When x values have equal weights, and there are no missing entries, up to 100 x values may be used.

Mathematical Basis of the Method

The procedure is a straightforward generalization of the curve-fitting case. Let z_{ij} be given for the point (x_i, y_j) with

$$i = 1(1)m, \quad j = 1(1)n,$$

and let orthogonal polynomials of degrees 0 to u in x and 0 to v in y be defined by the relations

$$\sum_{i=1}^{m} \phi_r(x_i)\phi_s(x_i) = 0 \quad r \neq s, \quad \sum_{i=1}^{m} \phi_r^2(x_i) = D_r \quad (2)$$

$$\sum_{j=1}^{n} \psi_r(y_j)\psi_s(y_j) = 0 \quad r \neq s, \quad \sum_{j=1}^{n} \psi_r^2(y_j) = \delta_r. \quad (3)$$

The polynomials for x are determined from the relations

$$\phi_{n+1}(x) = (x - \alpha_n)\phi_n(x) - \beta_n\phi_{n-1}(x),$$

where

$$\alpha_n = \sum_{i=1}^{m} x_i\phi_n^2(x_i)/D_n, \quad \beta_n = D_n/D_{n-1},$$

and initially

$$\phi_0(x) = 1, \quad \phi_1(x) = x - \bar{x}.$$

In practice, defining a suitable $\phi_{-1}(x)$ leads to a more compact program. Similar formulae lead to the y polynomials.

The first step consists in finding least squares curves for the n rows of x values.

$$g_j(x) = \sum_{r=0}^{u} \lambda_{rj}\phi_r(x) \qquad j = 1(1)n.$$

The coefficients are chosen so as to minimize

$$\sum_{i=1}^{m} \{g_j(x_i) - z_{ij}\}^2,$$

and by virtue of the relationships (2), they are given by

$$\lambda_{rj} = \sum_{i=1}^{m} z_{ij}\phi_r(x_i)/D_r.$$

Next the $(u + 1)$ rows

$$\lambda_{r1}, \lambda_{r2}, \ldots \lambda_{rn} \qquad r = 0(1)u$$

are expressed as polynomials in y

$$\sum_{s=0}^{r} \gamma_{rs}\psi_s(y) \qquad r = 0(1)u.$$

We obtain the results

$$\gamma_{rs} = \sum_{j=1}^{n} \lambda_{rj}\psi_s(y_j)/\delta_s$$

$$f(x, y) = \sum_{r=0}^{u} \sum_{s=0}^{v} \gamma_{rs}\phi_r(x)\psi_s(y). \quad (4)$$

It follows from (2) and (3) that the minimum sum of squares of residuals is

$$R_0 = \sum_{i=1}^{m} \sum_{j=1}^{n} \{z_{ij} - f(x_i, y_j)\}^2$$
$$= \sum_{i=1}^{m} \sum_{j=1}^{n} z_{ij}^2 - \sum_{r=0}^{u} \sum_{s=0}^{r} \gamma_{rs}^2 D_r\delta_s.$$

Moreover, the least squares solution with γ_{pq} omitted is obtained from (4) by omitting this term in the summation. The increased value of the sum of squares of residuals is

$$R = R_0 + \gamma_{pq}^2 D_p\delta_q.$$

This additive process applies for each term omitted in arriving at the fitted function required. Provided that the $j(i)$ values mentioned earlier satisfy (1), fitting up to given degrees in orthogonal form is equivalent to the same choice of degrees in standard polynomial form. The final stage of the process consists in reducing (4) to

$$f(x, y) = \sum_{i=0}^{t} \sum_{j=0}^{j(i)} a_{ij}x^iy^j.$$

The above analysis has assumed equal weights, otherwise these are specified by

$$w_i \text{ for } i = 1(1)m, \quad \text{and } w_j' \text{ for } j = 1(1)n$$

for x and y values respectively; they appear as multiplicative factors inside each summation over i and j. In particular we note that R_0 is now defined as

$$R_0 = \sum_{i=1}^{m} \sum_{j=1}^{n} w_iw_j'\{z_{ij} - f(x_i, y_j)\}^2,$$

and this must be borne in mind when considering its value. From the statistical point of view, if we have

$$\text{var}\,(z_{ij}) = \sigma^2/w_iw_j',$$

then σ^2 is estimated by dividing R by $(mn - k)$, where a total of k coefficients have been fitted.

References

Clenshaw, C. W. (1960). "Curve Fitting with a Digital Computer," *The Computer Journal*, Vol. 2, p. 170.

DeLury, D. B. (1950). *Values and Integrals of the Orthogonal Polynomials up to n = 26.* Toronto: University of Toronto Press.

Fisher, R. A., and Yates, F. (1957). *Statistical Tables for Biological, Agricultural and Medical Research*, 5th edition. Edinburgh: Oliver and Boyd.

Forsythe, G. E. (1957). "Generation and use of Orthogonal Polynomials for Data Fitting with a Digital Computer," *J. Soc. for Indust. and Appl. Math.*, Vol. 5, p. 74.

Kendall, M. G. (1951). *The Advanced Theory of Statistics*, Vol. 2, 3rd edition. London: Griffin.