

5.2 Methods of Curve Fitting

Data Linearization Method for $y = Ce^{Ax}$

Suppose that we are given the points $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ and want to fit an exponential curve of the form

$$(1) \quad y = Ce^{Ax}.$$

The first step is to take the logarithm of both sides:

$$(2) \quad \ln(y) = Ax + \ln(C).$$

Then introduce the change of variables:

$$(3) \quad Y = \ln(y), \quad X = x, \quad \text{and} \quad B = \ln(C).$$

This results in a linear relation between the new variables X and Y :

$$(4) \quad Y = AX + B.$$

The original points (x_k, y_k) in the xy -plane are transformed into the points $(X_k, Y_k) = (x_k, \ln(y_k))$ in the XY -plane. This process is called **data linearization**. Then the least-squares line (4) is fit to the points $\{(X_k, Y_k)\}$. The normal equations for finding A and B are

$$(5) \quad \begin{aligned} \left(\sum_{k=1}^N X_k^2 \right) A + \left(\sum_{k=1}^N X_k \right) B &= \sum_{k=1}^N X_k Y_k, \\ \left(\sum_{k=1}^N X_k \right) A + NB &= \sum_{k=1}^N Y_k. \end{aligned}$$

After A and B have been found, the parameter C in equation (1) is computed:

$$(6) \quad C = e^B.$$

Example 5.4. Use the data linearization method and find the exponential fit $y = Ce^{Ax}$ for the five data points $(0, 1.5), (1, 2.5), (2, 3.5), (3, 5.0),$ and $(4, 7.5)$.

Apply the transformation (3) to the original points and obtain

$$(7) \quad \begin{aligned} \{(X_k, Y_k)\} &= \{(0, \ln(1.5)), (1, \ln(2.5)), (2, \ln(3.5)), (3, \ln(5.0)), (4, \ln(7.5))\} \\ &= \{(0, 0.40547), (1, 0.91629), (2, 1.25276), (3, 1.60944), (4, 2.01490)\}. \end{aligned}$$

These transformed points are shown in Figure 5.4 and exhibit a linearized form. The equation of the least-squares line $Y = AX + B$ for the points (7) in Figure 5.4 is

$$(8) \quad Y = 0.391202X + 0.457367.$$

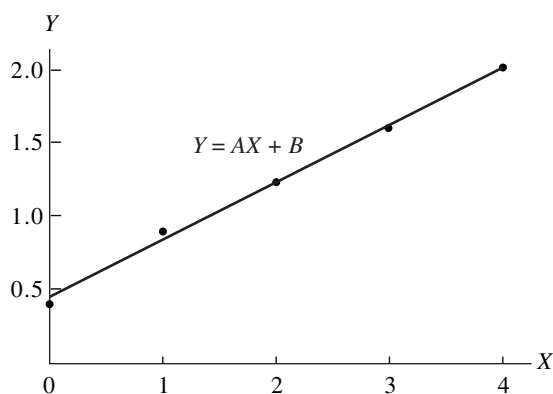


Figure 5.4 The transformed data points $\{(X_k, Y_k)\}$.

Table 5.4 Obtaining Coefficients of the Normal Equations for the Transformed Data Points $\{(X_k, Y_k)\}$

x_k	y_k	X_k	$Y_k = \ln(y_k)$	X_k^2	$X_k Y_k$
0.0	1.5	0.0	0.405465	0.0	0.000000
1.0	2.5	1.0	0.916291	1.0	0.916291
2.0	3.5	2.0	1.252763	4.0	2.505526
3.0	5.0	3.0	1.609438	9.0	4.828314
4.0	7.5	4.0	2.014903	16.0	8.059612
		10.0 $= \sum X_k$	6.198860 $= \sum Y_k$	30.0 $= \sum X_k^2$	16.309743 $= \sum X_k Y_k$

Calculation of the coefficients for the normal equations in (5) is shown in Table 5.4.

The resulting linear system (5) for determining A and B is

$$(9) \quad \begin{aligned} 30A + 10B &= 16.309742 \\ 10A + 5B &= 6.198860. \end{aligned}$$

The solution is $A = 0.3912023$ and $B = 0.457367$. Then C is obtained with the calculation $C = e^{0.457367} = 1.579910$, and these values for A and C are substituted into equation (1) to obtain the exponential fit (see Figure 5.5):

$$(10) \quad y = 1.579910e^{0.3912023x} \quad (\text{fit by data linearization}). \quad \blacksquare$$

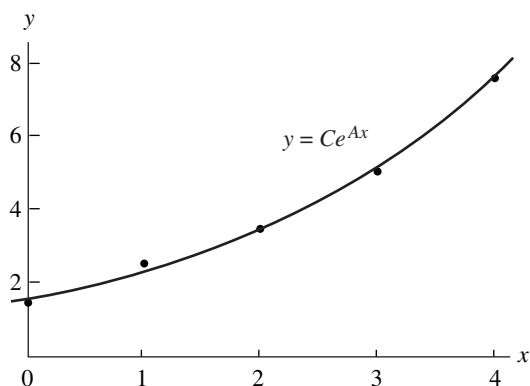


Figure 5.5 The exponential fit $y = 1.579910e^{0.3912023x}$ obtained by using the data linearization method.

Nonlinear Least-Squares Method for $y = Ce^{Ax}$

Suppose that we are given the points $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ and want to fit an exponential curve:

$$(11) \quad y = Ce^{Ax}.$$

The nonlinear least-squares procedure requires that we find a minimum of

$$(12) \quad E(A, C) = \sum_{k=1}^N (Ce^{Ax_k} - y_k)^2.$$

The partial derivatives of $E(A, C)$ with respect to A and C are

$$(13) \quad \frac{\partial E}{\partial A} = 2 \sum_{k=1}^N (Ce^{Ax_k} - y_k)(Cx_k e^{Ax_k})$$

and

$$(14) \quad \frac{\partial E}{\partial C} = 2 \sum_{k=1}^N (Ce^{Ax_k} - y_k)(e^{Ax_k}).$$

When the partial derivatives in (13) and (14) are set equal to zero and then simplified, the resulting normal equations are

$$(15) \quad \begin{aligned} C \sum_{k=1}^N x_k e^{2Ax_k} - \sum_{k=1}^N x_k y_k e^{Ax_k} &= 0, \\ C \sum_{k=1}^N e^{Ax_k} - \sum_{k=1}^N y_k e^{Ax_k} &= 0. \end{aligned}$$

The equations in (15) are nonlinear in the unknowns A and C and can be solved using Newton's method. This is a time-consuming computation and the iteration involved requires good starting values for A and C . Many software packages have a built-in minimization subroutine for functions of several variables that can be used to minimize $E(A, C)$ directly. For example, the Nelder-Mead simplex algorithm can be used to minimize (12) directly and bypass the need for equations (13) through (15).

Example 5.5. Use the least-squares method and determine the exponential fit $y = Ce^{Ax}$ for the five data points $(0, 1.5)$, $(1, 2.5)$, $(2, 3.5)$, $(3, 5.0)$, and $(4, 7.5)$.

For this solution we must minimize the quantity $E(A, C)$, which is

$$(16) \quad E(A, C) = (C - 1.5)^2 + (Ce^A - 2.5)^2 + (Ce^{2A} - 3.5)^2 \\ + (Ce^{3A} - 5.0)^2 + (Ce^{4A} - 7.5)^2.$$

We use the `fmins` command in MATLAB to approximate the values of A and C that minimize $E(A, C)$. First we define $E(A, C)$ as an M-file in MATLAB.

```
function z=E(u)
A=u(1);
C=u(2);
z=(C-1.5).^2+(C.*exp(A)-2.5).^2+(C.*exp(2*A)-3.5).^2+...
(C.*exp(3*A)-5.0).^2+(C.*exp(4*A)-7.5).^2;
```

Using the `fmins` command in the MATLAB Command Window and the initial values $A = 1.0$ and $C = 1.0$, we find

```
>>fmins('E',[1 1])
ans =
0.38357046980073 1.61089952247928
```

Thus the exponential fit to the five data points is

$$(17) \quad y = 1.6108995e^{0.3835705x} \quad (\text{fit by nonlinear least squares}).$$

A comparison of the solutions using data linearization and nonlinear least squares is given in Table 5.5. There is a slight difference in the coefficients. For the purpose of interpolation it can be seen that the approximations differ by no more than 2% over the interval $[0, 4]$ (see Table 5.5 and Figure 5.6). If there is a normal distribution of the errors in the data, (17) is usually the preferred choice. When extrapolation is made beyond the range of the data, the two solutions will diverge and the discrepancy increases to about 6% when $x = 10$. ■

Numerical Methods Using Matlab, 4th Edition, 2004

John H. Mathews and Kurtis K. Fink

ISBN: 0-13-065248-2

Prentice-Hall Inc.

Upper Saddle River, New Jersey, USA

<http://vig.prenhall.com/>

NUMERICAL METHODS USING MATLAB

FOURTH EDITION



JOHN H. MATHEWS • KURTIS D. FINK