Least-squares fitting of parametric curves with a linear function of several variables as argument

HELMUTH SPÄTH[*]

Abstract. *We discuss fitting of a parametric curve in the plane in the least-squares sense when the independent variable is a linear function of several variables with unknown coefficients. A general numerical method is recommended. For two special models the algorithmic details and numerical examples are given.*

Key words: *orthogonal least squares, TLS*

Sažetak. **Fitovanje u smislu najmanih kvadrata parametarskih krivulja kojima je nezavisna varijabla linearna funkcija više varijabli**. *U radu se diskutira fitovanje u smislu najmanjih kvadrata parametarski zadane ravninske krivulje pri čemu je nezavisna varijabla linearna funkcija više varijabli s nepoznatim koeficijentima. Predlaže se opća numerička metoda. Za dva specijalna modela navedeni su algoritamski detalji i numerički primjeri.*

Ključne riječi: *ortogonalna metoda najmanjih kvadrata, potpuna metoda najmanjih kvadrata*

AMS subject classifications: 65D10

Received February 9, 1998 Accepted March 11, 1998

1. The general problem

Let

$$\begin{aligned} \boldsymbol{a} &= (a_1,\ldots,a_u)^T\,, \\ \boldsymbol{b} &= (b_1,\ldots,b_v)^T\,, \\ \boldsymbol{c} &= (c_1,\ldots,c_n)^T \end{aligned} \tag{1}$$

denote parameters of some curve to be estimated. The curve is assumed to be given by

$$\begin{aligned} x &= f(\boldsymbol{a};z)\,, \\ y &= g(\boldsymbol{b};z)\,, \end{aligned} \tag{2}$$

[*]Fachbereich Mathematik, Carl von Ossietzky Universität Oldenburg, Postfach 2503, 26111 Oldenburg, Germany, e-mail: `spaeth@mathematik.uni-oldenburg.de`

where the independent variable z is some linear function of $n \geq 2$ variables $t_1, \dots, t_n$, i.e.

$$z = c_1 t_1 + \cdots + c_n t_n \,. \tag{3}$$

Further, let the measured points

$$(x_j, y_j), \quad j = 1, \dots, m > u + v + n \tag{4}$$

be given in the plane and measured values

$$(t_{kj}), \quad k = 1, \dots, n,\ j = 1, \dots, m \tag{5}$$

for the variables t_k, $k = 1, \dots, n$, too.

We want to fit the parameters $\boldsymbol{a}, \boldsymbol{b}$, and $\boldsymbol{c}$ such that

$$S(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) = \sum_{j=1}^{m} \big(x_j - f(\boldsymbol{a}; z_j)\big)^2 + \big(y_j - g(\boldsymbol{b}; z_j)\big)^2 \tag{6}$$

is minimized, where

$$z_j = z_j(\boldsymbol{c}) = c_1 t_{1j} + \cdots + c_n t_{nj}, \quad j = 1, \dots, m \tag{7}$$

corresponding to (3).

In turn, we consider for each parameter set the necessary conditions for a minimum of (6). We have

$$\frac{\partial S}{\partial a_i} = 0 \quad \Longleftrightarrow \quad \sum_{j=1}^{m} \frac{\partial f}{\partial a_i}(\boldsymbol{a}; z_j)\big(x_j - f(\boldsymbol{a}; z_j)\big) = 0, \quad i = 1, \dots, u \,, \tag{8}$$

$$\frac{\partial S}{\partial b_k} = 0 \quad \Longleftrightarrow \quad \sum_{j=1}^{m} \frac{\partial g}{\partial b_k}(\boldsymbol{b}; z_j)\big(y_j - g(\boldsymbol{b}; z_j)\big) = 0, \quad k = 1, \dots, v \,. \tag{9}$$

If we now assume – this is true in most practical models of type (2) – that $\boldsymbol{a}$ and $\boldsymbol{b}$ linearly appear in (2), then (8) and (9) are *linear* systems of equations for u variables $\boldsymbol{a}$ and v variables $\boldsymbol{b}$ if $\boldsymbol{c}$, i.e. z_j, $j = 1, \dots, m$ is *fixed*. Finally,

$$\begin{aligned} \frac{\partial S}{\partial c_r} = 0 \quad \Longleftrightarrow \quad & \sum_{j=1}^{m} t_{rj} \bigg[\frac{\partial f}{\partial z}(\boldsymbol{a}; z_j)\big(x_j - f(\boldsymbol{a}; z_j)\big) \\ & + \frac{\partial g}{\partial z}(\boldsymbol{b}; z_j)\big(y_j - g(\boldsymbol{b}; t_j)\big) \bigg] = 0 \,, \quad r = 1, \dots, n \,. \end{aligned} \tag{10}$$

For *given* $\boldsymbol{a}$ and $\boldsymbol{b}$ this is a *nonlinear* system of n equations for n unknowns $\boldsymbol{c}$.

The structure of the equations (8), (9), and (10) suggests the following algorithm indicated by Mardia for angular regression (see [1]):

Step 0. Let the starting values $\boldsymbol{c}^{(0)}$ for $\boldsymbol{c}$ be given. Set $\ell = 0$.

Step 1. Solve the linear systems (8) and (9) for $\boldsymbol{c} = \boldsymbol{c}^{(\ell)}$ and set $\boldsymbol{a}^{(\ell)} = \boldsymbol{a}$, $\boldsymbol{b}^{(\ell)} = \boldsymbol{b}$.

Step 2. For $\boldsymbol{a} = \boldsymbol{a}^{(\ell)}$, $\boldsymbol{b} = \boldsymbol{b}^{(\ell)}$ perform one (or several) Newton iterations for (10), i.e. set

$$\boldsymbol{c}^{(\ell+1)} = \boldsymbol{c}^{(\ell)} - F'\big(\boldsymbol{c}^{(\ell)}\big)^{-1} F\big(\boldsymbol{c}^{(\ell)}\big) \tag{11}$$

and, if no convergence has occurred, set $\ell := \ell + 1$ and go back to Step 1.

In (11) we have

$$F(\boldsymbol{c}) = \nabla S(\boldsymbol{c}) = \left(\frac{\partial S}{\partial c_1}(\boldsymbol{c}), \dots, \frac{\partial S}{\partial c_n}(\boldsymbol{c})\right) \tag{12}$$

(see (10)), and

$$F'(\boldsymbol{c}) = \nabla^2 S(\boldsymbol{c}) = \left(\frac{\partial^2 S}{\partial c_r \partial c_s}\right)_{r,s=1,\dots,n} \tag{13}$$

is the Jacobian with

$$\begin{aligned}\frac{\partial^2 S}{\partial c_r \partial c_s} = -\sum_{j=1}^{m} t_{rj}\, t_{sj} \Bigg[& \frac{\partial^2 f}{\partial z^2}(\boldsymbol{a}; z_j)\big(x_j - f(\boldsymbol{a}; z_j)\big) - \left(\frac{\partial f}{\partial z}(\boldsymbol{a}; z_j)\right)^2 \\ & + \frac{\partial^2 g}{\partial z^2}(\boldsymbol{b}; z_j)\big(y_j - g(\boldsymbol{b}; z_j)\big) - \left(\frac{\partial g}{\partial z}(\boldsymbol{b}; z_j)\right)^2 \Bigg].\end{aligned} \tag{14}$$

In the next two sections we will discuss the algorithm for two special models (2) and we will give numerical examples and corresponding experiences with the above algorithm.

2. The straight line

Without loss of generality, a straight line can be parametrized by

$$\begin{aligned} x &= a_1 + a_2 z\,, \\ y &= b_1 + z\,. \end{aligned} \tag{15}$$

Then, $u = 2$, $v = 1$, and

$$S(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) = \sum_{j=1}^{m} (x_j - a_1 - a_2 z_j)^2 + (y_j - b_1 - z_j)^2\,. \tag{16}$$

Here

$$\frac{\partial S}{\partial a_1} = \frac{\partial S}{\partial a_2} = 0\,, \quad \frac{\partial S}{\partial b_1} = 0$$

can very easily be solved for fixed $\boldsymbol{c}$ to give a_1, a_2, and b_1. The necessary conditions (10) for $\boldsymbol{c}$ are

$$\sum_{j=1}^{m} t_{rj}(c_1 t_{1j} + \dots + c_n t_{nj}) = \tfrac{1}{a_2^2+1} \sum_{j=1}^{m} t_{rj} \big[a_2(x_j - a_1) + (y_j - b_1)\big]\,, \quad r = 1, \dots, n\,. \tag{17}$$

In this special case thus (10) is a *linear* system for $\boldsymbol{c}$. One Newton iteration in Step 2 of our algorithm means just solving (17). The numerical solution of (17) is preferably realized by using the QR decomposition (see [2] and subroutine MGS in [3]) to solve the overdetermined system

$$\begin{pmatrix} t_{11} & t_{21} & \dots & t_{n1} \\ t_{12} & t_{22} & \dots & t_{n2} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ t_{1m} & t_{2m} & \dots & t_{nm} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_m \end{pmatrix}, \tag{18}$$

where

$$d_j = \frac{a_2(x_j - a_1) + (y_j - b_1)}{a_2^2 + 1}, \quad j = 1, \dots, m, \tag{19}$$

in the least-squares sense. Note that the matrix (t_{kj}) does not change during the iteration, i.e. the QR decomposition can be made once for all. Only the right-hand side d is changed because $d = d(\boldsymbol{a}, \boldsymbol{b})$.

Example 1. *The data were generated using* $a_1 = 0$, $a_2 = 1/2$, $b_1 = 1$, $c_1 = 6$, $c_2 = 8$, $m = 7$, $n = 2$ *and*

t_{1j}	0	1	1	1	−1	2	1
t_{2j}	1	0	1	−1	1	1	−2

to give

x_j	4	3	7	−1	1	10	−5
y_j	9	7	15	−1	3	21	−9

Then, we disturbed t_{1j}, t_{2j}, x_j, y_j *into*

t_{1j}	0	0.9	1	0.8	−1	2.1	1
t_{2j}	1	0	0.9	−1	1.1	0.9	−2
x_j	4	4	7	−2	1	11	−5
y_j	8	7	15	−1	4	22	−8

For three different starting values $\mathbf{c}^{(0)} = (6, 8)$, $\mathbf{c}^{(0)} = (2, -4)$, *and* $\mathbf{c}^{(0)} = (-10, 10)$ *we got within 7 iterations up to 4 decimal digit accuracy* $a_1 = -.099$, $a_2 = 0.545$, $b_1 = 1.286$, $c_1 = 6.458$, $c_2 = 7.776$, *and* $S = 2.939$ *in each case. The results are very likely to represent the absolute minimum.*

Example 2. *For the completely arbitrary data*

t_{1j}	1	5	2	−3	0	3	−2	0
t_{2j}	2	3	0	4	−2	−7	5	0
x_j	3	1	−4	9	1	−4	−3	0
y_j	4	−2	−3	0	4	0	2	0

and for the same three starting values as above we got within 8 iterations $a_1 = 0.641$, $a_2 = 3.078$, $b_1 = 0.711$, $c_1 = -0.197$, $c_2 = 0.098$ *and* $S = 136.18$. *This might not correspond to the absolute minimum here.*

3. The ellipse in normal position

We use the parametrization

$$\begin{aligned} x &= a + p\cos z\,, \\ y &= b + q\sin z\,. \end{aligned} \tag{20}$$

Here (a, b) is the center and (p, q) are the half axes. We have $u = v = 2$, $(a_1, a_2) = (a, p)$, $(b_1, b_2) = (b, q)$. The function to be minimized is

$$S(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) = \sum_{j=1}^{m} (x_j - a - p\cos z_j)^2 + (y_j - b - q\sin z_j)^2\,, \tag{21}$$

where $z_j = z_j(\boldsymbol{c})$ is given by (7). Note with $(a, b, p, q, \boldsymbol{c})$ also $(a, b, p, -q, -\boldsymbol{c})$ would be a solution. For (8) and (9) we have

$$\begin{aligned} ma \quad &+ \quad p\sum_{j=1}^{m} \cos z_j \quad = \quad \sum_{j=1}^{m} x_j\,, \\ a\sum_{j=1}^{m} \cos z_j \quad &+ \quad p\sum_{j=1}^{m} \cos^2 z_j \quad = \quad \sum_{j=1}^{m} x_j \cos z_j\,, \end{aligned} \tag{22}$$

and

$$\begin{aligned} mb \quad &+ \quad q\sum_{j=1}^{m} \sin z_j \quad = \quad \sum_{j=1}^{m} y_j\,, \\ b\sum_{j=1}^{m} \sin z_j \quad &+ \quad q\sum_{j=1}^{m} \sin^2 z_j \quad = \quad \sum_{j=1}^{m} y_j \sin z_j\,. \end{aligned} \tag{23}$$

These two 2×2 systems can very easily be solved. However, (10) now results into nonlinear equations

$$\begin{aligned} \sum_{j=1}^{m} t_{rj}\big[(q^2 - p^2)\sin z_j \cos z_j + p\sin z_j (x_j - a) - q\cos z_j (y_j - b)\big] = 0\,, \\ r = 1, \ldots, n\,, \end{aligned} \tag{24}$$

and (14) gives

$$\frac{\partial^2 S}{\partial c_r \partial c_s} = \sum_{j=1}^{m} t_{rj} t_{sj} \big[(q^2 - p^2)(\cos^2 z_j - \sin^2 z_j) + p\cos z_j (x_j - a) + q\sin z_j (y_k - b)\big]. \tag{25}$$

Thus, the Newton step can easily be implemented.

Example 3. *The data*

t_{1j}	−3.2	−0.9	2.8	−1.3	0.8	7.1	−0.9	−3	7
t_{2j}	−5	−3	−1	1.2	2.8	3	5.2	7.3	9.1
x_j	1	−2	4	3	−2	−2	−1	3	−2
y_j	3	−4	−4	2	0	−3	−6	−5	1

were first generated and then disturbed like in Example 1. *It was necessary now to use far more starting values* $\boldsymbol{c}^{(0)}$ *to get an acceptable minimum. We generated* *one hundred values for* $\mathbf{c}^{(0)}$ *by choosing the components of this vector randomly and equally distributed in* $[-1, 1]$. *The smallest value for* S *was* 3.045. *It appeared in about 8 iterations in 11 out of one hundred cases. The corresponding parameters were* $a = 0.726$, $b = -1.920$, $p = 3.274$, $q = 4.596$, $c_1 = 0.148$, $c_2 = 0.840$. *A figure to be made indicates that these values correspond to the absolute minimum. We recommend to use a large number of starting values when* f *and* g *are* nonlinear *with respect to* z. *Nevertheless, the computing time for* Example 3 *was negligible on a PC.*

References

[1] K. V. MARDIA, *Statistics of Directional Data*, Academic Press, 1972.

[2] G. H. GOLUB, C. F. VAN LOAN, *Matrix Computations*, 3rd edition, John Hopkins University Press, 1996.

[3] H. SPÄTH, *Numerik – Eine Einführung für Mathematiker und Informatiker*, Vieweg, 1994.