NLR

NLR produces the least square estimates of the parameters for models that are not linear in their parameters. Unlike in other procedures, the weight function is not treated as a case replicate in NLR.

Model

Consider the model

$$f = f(\mathbf{x}, \Theta)$$

where Θ is a $p \times 1$ parameter vector, $\mathbf{x}$ is a vector of independent variables, and f is a function of $\mathbf{x}$ and Θ .

Goal

Find the least square estimate Θ^* of Θ such that Θ^* minimizes the objective function

$$F(\Theta) = \mathbf{R'WR} \qquad \textbf{(1)}$$

where

$$\begin{aligned}
\mathbf{R'} &= (R_1, \ldots, R_n) \\
R_i &= y_i - f_i \\
f_i &= f(x_i, \Theta^*), \quad i = 1, \ldots, n \\
\mathbf{W} &= \text{Diag}(W_1, \ldots, W_n)
\end{aligned}$$

and n is the number of cases. For case i, y_i is the observed dependent variable, x_i is the vector of observed independent variables, W_i is the weight function which can be a function of Θ .

The gradient of F at Θ_j is defined as

$$\nabla F = 2\mathbf{J}'_{\cdot j}\mathbf{WR}$$

where $\mathbf{J}_{\cdot j}$ is the jth column of the $n \times p$ Jacobian matrix $\mathbf{J}$ whose (i, j) th element J_{ij} is defined by

$$J_{ij} = \frac{R_i}{2W_i}\frac{\partial W_i}{\partial \Theta_j} - \frac{\partial f_i}{\partial \Theta_j}$$

Estimation

The modified Levenberg-Marquardt algorithm that was proposed by Moré (1977) and is contained in MINPACK is used in NLR to solve equation **(1)**.

Given an initial value $\Theta^{(0)}$ for Θ, the algorithm is as follows:

At stage $k+1, k = 0, 1, 2, \ldots$

- Compute

$$f_i^{(k)} = f_i\left(\Theta^{(F)}\right),\ R_i^{(k)} = y_i - f_i^{(k)},\ F_k = F\left(\Theta^{(k)}\right), \text{ and } J^{(k)} = J\left(\Theta^{(k)}\right)$$

- Choose an appropriate non-negative scalar such that

$$F\left(\Theta^k + h_k\right) < F_k$$

where

$$h_k = -\left(\mathbf{J}^{(k)'}\mathbf{J}^{(k)} + \alpha_k \mathbf{I}\right)^{-1}\mathbf{J}^{(k)'}\mathbf{R}^{(k)}$$

- Set

$$\Theta^{(k+1)} = \Theta^{(k)} + h_k$$

and compute $\mathbf{J}^{(k+1)}, \mathbf{R}^{(k+1)}, \mathbf{W}^{(k+1)}$, and F_{k+1}.

- Check the following conditions:

(1) $1 - \left(F_{k-1} / F_k\right) < \varepsilon_1 (SSCON)$

(2) For every element of h_k,

$$\left| h_{ki} \big/ \Theta_i^{(k)} \right| < \varepsilon_2 (PCON)$$

(3) $k + 1 \geq ITER$ (maximum number of iterations)

(4) For every parameter Θ_j, the gradient of F at Θ_j, ∇F_j, is evaluated at $\Theta^{(k+1)}$ by checking

$$\left| r_j^{(k+1)} \right| < \varepsilon_2 (RCON)$$

where $r_j^{(k+1)}$ is the correlation between the *j*th column $\mathbf{J}_j^{(k+1)}$ of $\mathbf{J}^{(k+1)}$ and $\mathbf{W}^{(k+1)}\mathbf{R}^{(k+1)}$.

If any of these four conditions is satisfied, the algorithm will stop. Then the final parameter estimate Θ^*

$$\Theta^* = \Theta^{(k+1)}$$

and the termination reason is reported. Otherwise, iteration continues.

Statistics

When iteration terminates, the following statistics are printed.

Parameter Estimates and Standard Errors

The asymptotic standard error of Θ_j^* is estimated by the square root of the jth diagonal element a_{jj} of **A**, where

$$\mathbf{A} = \frac{F\left(\Theta^*\right)}{n-p}\left(\mathbf{J}^{*\prime}\mathbf{W}^*\mathbf{J}^*\right)^{-1}$$

and $\mathbf{J}^*$ and $\mathbf{W}^*$ are the Jacobian matrix **J** and weight function **W** evaluated at Θ^*, respectively.

Asymptotic 95% Confidence Interval for Θ_j

$$\Theta_j^* \pm t(0.975, n-p)a_{ii}$$

Asymptotic Correlation Matrix of the Parameter Estimates

$$\mathbf{C} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$$

where

$$\mathbf{D} = \text{Diag}\left(a_{11,\ldots,}\ a_{pp}\right)$$

and a_{ii} is the ith diagonal element of A.

Analysis of Variance Table

Source	**df**	**Sum of Squares**
Residual	$n-p$	$F\left(\Theta^*\right)$
Regression	p	$SS_{\text{uncorrected}} - F\left(\Theta^*\right)$
Uncorrected Total	n	$SS_{\text{uncorrected}}$
Corrected Total	$n-1$	$SS_{\text{uncorrected}} - \bar{y}^2 \sum_{i=1}^{n} W_i\left(\Theta^*\right)$

where

$$SS_{\text{uncorrected}} = \sum_{i=1}^{n} W_i\left(\mathrm{Q}^*\right) y_i^2$$

$$\bar{y} = \left(\sum_{i=1}^{n} W_i\left(\mathrm{Q}^*\right) y_i\right) \Bigg/ \left(\sum_{i=1}^{n} W_i\left(\mathrm{Q}^*\right)\right)$$

References

Moré, J. J. 1977. The Levenberg-Marquardt algorithm: implementation and theory in numerical analysis. In: *Lecture Notes in Mathematics* 630, G. A. Watson, ed. Berlin: Springer-Verlag. 105–116.