1   Introduction

In this section, we examine regression models that are nonlinear in the parameters and give a brief overview of methods to estimate such models.

2   Nonlinear Regression Models

The general form of the nonlinear regression model is

$$y_i = h(x_i, \beta, \epsilon_i), \tag{1}$$

which is more commonly written in a form with an additive error term

$$y_i = h(x_i, \beta) + \epsilon_i. \tag{2}$$

Below are two examples

1. $h(x_i, \beta, \epsilon_i) = \beta_0 x_{1i}^{\beta_1} x_{2i}^{\beta_2} \exp(\epsilon_i)$.   This is an <u>intrinsically linear</u> model because by taking natural logarithms, we get a model that is linear in the parameters, $\ln(y_i) = \beta_0 + \beta_1 \ln(x_{1i}) + \beta_2 \ln(x_{2i}) + \epsilon_i$.   This can be estimated with standard linear procedures such as OLS.

2. $h(x_i, \beta) = \beta_0 x_{1i}^{\beta_1} x_{2i}^{\beta_2}$.   Since the error term in (2) is additive, there is no transformation that will produce a linear model.   This is an intrinsically nonlinear model (i.e., the relevant first-order conditions are nonlinear in the parameters).   Below we consider two methods for estimating such a model – linearizing the underlying regression model and nonlinear optimization of the objective function.

2.1   Linearized Regression Model and the Gauss-Newton Algorithm

Consider a first-order Taylor series approximation of the regression model around β_0

$$y_i = h(x_i, \beta) + \epsilon_i \simeq h(x_i, \beta_0) + g(x_i, \beta_0)(\beta - \beta_0) + \epsilon_i$$

where

$$g(x_i, \beta_0) = (\partial h / \partial \beta_1 |_{\beta = \beta_0}, ..., \partial h / \partial \beta_k |_{\beta = \beta_0}).$$

Collecting terms and rearranging gives

$$Y^0 = X^0\beta + \epsilon^0$$

where

$$
\begin{aligned}
Y^0 &\equiv Y - h(X, \beta_0) + g(X, \beta_0)\beta_0 \\
X^0 &\equiv g(X, \beta_0).
\end{aligned}
$$

The matrix X^0 is called the pseudoregressor matrix. Note also that ϵ^0 will include higher-order approximation errors.

2.1.1 Gauss-Newton Algorithm

Given an initial value for β_0, we can estimate β with the following iterative LS procedure

$$
\begin{aligned}
b_{t+1} &= [X^0(b_t)'X^0(b_t)]^{-1}[X^0(b_t)'Y^0(b_t)] \\
&= [X^0(b_t)'X^0(b_t)]^{-1}[X^0(b_t)'(X^0(b_t)b_t + e_t^0)] \\
&= b_t + [X^0(b_t)'X^0(b_t)]^{-1}X^0(b_t)'e_t^0 \\
&= b_t + W_t\lambda_t g_t
\end{aligned}
$$

where $W_t = [2X^0(b_t)'X^0(b_t)]^{-1}$, $\lambda_t = 1$ and $g_t = 2X^0(b_t)'e_t^0$. The iterations continue until the difference between b_{t+1} and b_t is sufficiently small. This is called the Gauss-Newton algorithm. Interpretations for W_t, λ_t and g_t will be given below. A consistent estimator of σ^2 is

$$s^2 = \frac{1}{n-k}\sum_{i=1}^{n}(y_i - h(x_i, b))^2.$$

2.1.2 Properties of the NLS Estimator

Only asymptotic results are available for this estimator. Assuming that the pseudoregessors are well-behaved (i.e., $\text{plim}\frac{1}{n}X^{0\prime}X^0 = Q^0$, a finite positive definite matrix), then we can apply the CLT to show that

$$b \overset{asy}{\sim} N[\beta, \frac{\sigma^2}{n}(Q^0)^{-1}],$$

where the estimate of $\frac{\sigma^2}{n}(Q^0)^{-1}$ is $s^2(X^{0\prime}X^0)^{-1}$.

2.1.3 Notes

1. Depending on the initial value, b_0, the Gauss-Newton algorithm can lead to a local (as opposed to global) minimum or head off on a divergent path.

2. The standard R^2 formula may produce a goodness-of-fit value outside the interval $[0, 1]$.

3. Extensions of the J test are available that allow one to test nonlinear versus linear models.

4. Hypothesis testing is only valid asymptotically.

2.2 Hypothesis Testing

Consider testing the hypothesis H$_0$: $R(\beta) = q$. Below are four tests that are asymptotically equivalent.

2.2.1 Asymptotic F test.

Begin by letting $S(b) = (Y - h(X, b))'(Y - h(X, b))$ be the sum of square residuals evaluated at the unrestricted NLS estimate. Also, let $S(b_*)$ be the corresponding measure evaluated at the restricted estimate. The standard F formula gives

$$F = \frac{[S(b_*) - S(b)]/J}{S(b)/(n - k)}.$$

Under the null hypothesis, $JF \overset{asy}{\sim} \chi^2(J)$.

2.2.2 Wald Test.

The nonlinear counterpart to the Wald statistic is

$$W = [R(b) - q]'[C\hat{V}C']^{-1}[R(b) - q] \overset{asy}{\sim} \chi^2(J)$$

where $\hat{V} = \hat{\sigma}^2(X^{0\prime}X^0)^{-1}$, $C = \partial R(b)/\partial b$ and $\hat{\sigma}^2 = S(b)/n$.

2.2.3 Likelihood Ratio Test.

Assume $\epsilon \sim N(0, \sigma^2 I)$. The likelihood ratio statistic is

$$LR = -2[\ln(L_*) - \ln(L)] \overset{asy}{\sim} \chi^2(J)$$

where $\ln(L)$ and $\ln(L_*)$ are the unrestricted and restricted (log) likelihood values respectively.

2.2.4 Lagrange Multiplier Test.

The LM statistic is based solely on the restricted model. Occasionally, by imposing the restriction $R(\beta) = q$, it may change an intrinsically nonlinear model into an intrinsically linear one. The LM statistic is

$$
\begin{aligned}
LM \quad &= \quad \frac{e'_* X^0_* [X^{0\prime}_* X^0_*]^{-1} X^{0\prime}_* e_*}{S(b_*)/n} \\
&= \quad \frac{e'_* X^0_* \tilde{b}}{S(b_*)/n} = n \frac{\widetilde{ESS}}{\widetilde{TSS}} = n\tilde{R}^2 \overset{asy}{\sim} \chi^2(J)
\end{aligned}
$$

where $e_* = Y - h(X, b_*)$, $X^0_* = g(X, b_*)$, $\tilde{b}$ is the estimated coefficient of e_* on X^0_* and $\tilde{R}^2$ is the coefficient of determination of e_* on X^0_*.

3 Brief Overview of Nonlinear Optimization Techniques

An alternative method for estimating the parameters of equation (2) is to apply nonlinear optimization techniques directly to the first-order conditions. Consider the NLS problem of minimizing

$$
S(b) = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - h(x_i, b))^2.
$$

The first-order conditions produce

$$
\frac{\partial S(b)}{\partial b} = -2 \sum_{i=1}^{n} (y_i - h(x_i, b)) \frac{\partial h(x_i, b)}{\partial b} = 0, \tag{3}
$$

which is generally nonlinear in the parameters and does not have a nice closed-form, analytical solution. The methods outlined below can be used to solve the set of equations (3).

3.1 Introduction

Consider the function $f(\theta) = a + b\theta + c\theta^2$. The first-order condition for minimization is

$$
\frac{df(\theta)}{d\theta} = b + 2c\theta = 0 \Longrightarrow \theta^* = -b/2c.
$$

This is considered a linear optimization problem even though the objective function is nonlinear in the parameters. Alternatively, consider the objective fuction $f(\theta) = a + b\theta^2 + c\ln(\theta)$. The first-order condition for minimization is

$$
\frac{df(\theta)}{d\theta} = 2b\theta + \frac{c}{\theta} = 0.
$$

This is considered a nonlinear optimization problem.

Here is a general outline of how to solve the nonlinear optimization problem. Let θ be the parameter vector, Δ the directional vector and λ the step length.

Procedure.

1. Specify θ_0 and Δ_0.

2. Determine λ.

3. Compute $\theta_{t+1} = \theta_t + \lambda_t \Delta_t$.

4. Convergence criterion satisfied?

   - Yes $\longrightarrow$ Exit.

   - No $\longrightarrow$ Update $t = t + 1$, compute Δ_t and return to #2.

There are two general types of nonlinear optimization algorithms – those that do not involve derivatives and those that do.

3.2 Derivative-Free Methods

Derivative-free algorithms are used when the number of parameters are small, analytical derivatives are difficult to calculate or seed values are needed for other algorithms.

1. Grid search. This is a trial-and-error method that is typically not feasible for more than two parameters. It can be a useful means to find starting values for other algorithms.

2. Direct search methods. Using the iterative algorithm $\theta_{t+1} = \theta_t + \lambda_t \Delta_t$, a search is performed in m directions: $\Delta_1, ..., \Delta_m$. λ_t is chosen to ensure that $G(\theta_{t+1}) > G(\theta_t)$.

3. Other methods. Simplex algorithm and simulated annealing are examples of other derivative-free methods.

3.3 Gradient Methods

The goal is to choose a directional vector Δ_t to go uphill (for a max) and an appropriate step length λ_t. Too big a step may overshoot a max and too small a step may be inefficient. (See Figures 5.3 and 5.4 attached.) With this in mind, consider choosing λ_t such that the objective function increases (i.e., $G(\theta_{t+1}) > G(\theta_t)$). The relevant derivative is

$$\frac{dG(\theta_t + \lambda_t \Delta_t)}{d\lambda_t} = g_t' \Delta_t$$

where $g_t = dG(\theta_{t+1})/d\theta_{t+1}$. If we let $\Delta_t = W_t g_t$, where W_t is a positive definite matrix, then we know that

$$\frac{dG(\theta_t + \lambda_t \Delta_t)}{d\lambda_t} = g_t' W_t g_t \geq 0.$$

As a result, almost all algorithms take the general form

$$\theta_{t+1} = \theta_t + \lambda_t W_t g_t$$

where λ_t is the step length, W_t is a weighting matrix, and g_t is the gradient. The Gauss-Newton algorithm above could be written in this general form. Here are examples of some other algorithms.

1. Steepest Ascent.

   - $W_t = I$ so $\Delta_t = g_t$.

   - An optimal line search produces $\lambda_t = -g'g/(g'Hg)$.

   - Therefore, the algorithm is $\theta_{t+1} = \theta_t - [g_t'g_t/(g_t'H_tg_t)]g_t$.

   - This method has the drawbacks that (a) it can be slow to converge, especially on long narrow ridges and (b) H can be difficult to calculate.

2. Newton's Method (aka Newton-Raphson).

   - Newton's method can be motivated by taking a Taylor series approximation (around θ_0) of the gradient and setting equal to zero. This gives $g(\theta_t) \simeq g(\theta_0) + H(\theta_0)[\theta_t - \theta_0]$. Rearranging, produces $\theta_t = \theta_0 - H^{-1}(\theta_0)g(\theta_0)$.

   - Therefore, $W_t = -H^{-1}$ and $\lambda_t = 1$.

   - Very popular and works well in many settings.

   - Hessian can be difficult to calculate or positive definite if far from optimum.

   - Newton's method will reach the optimum in one step if $G(\theta_t)$ is quadratic.

3. Quadratic Hill Climbing.

   - $W_t = -(H(\theta_t) - \alpha I)^{-1}$, where $\alpha > 0$ is chosen to ensure that W_t is positive definite.

4. Davidson-Fletcher-Powell (DFP).

   - $W_{t+1} = W_t + E_t$, where E_t is a positive definite matrix.

   - $E_t = \frac{(\lambda_t \Delta_t)(\lambda_t \Delta_t)'}{(\lambda_t \Delta_t)(g_t - g_{t-1})} + \frac{W_t(g_t - g_{t-1})(g_t - g_{t-1})'W_t}{(g_t - g_{t-1})'W_t(g_t - g_{t-1})}$.

   - Notice that no second derivatives (i.e., $H(\theta_t)$) are required.

- Choose $W_0 = I$.

5. <u>Method of Scoring</u>.

   - $W_t = - \left(E\left(\frac{\partial^2 \ln(L)}{\partial\theta\partial\theta'}\right) \right)^{-1}$.

6. <u>BHHH or Outer Product of the Gradients</u>.

   - $W_t = (g(\theta_t)g(\theta_t)')^{-1}$ is an estimate of $-H^{-1}(\theta_t) = -(\frac{\partial^2 \ln(L)}{\partial\theta\partial\theta'})^{-1}$.

   - W_t is always positive definite.

   - Only requires first derviatives.

<u>Notes</u>.

1. Nonlinear optimization with constraints. There are several options such as forming a Lagrangian function, substituting the constraint directly into the objective and imposing arbitrary penalties into the objective function.

2. Assessing convergence.

   - The ususal choice of convergence criterion is G or θ.

   - Sometimes these methods can be sensitive to the scaling of the function.

   - Belsley suggests using $g'H^{-1}g$ as the criterion, which removes the units of measurement.

3. The biggest problem in nonlinear optimization is making sure the solution is a global, as opposed to local, optimum. The above methods work well for globally concave (convex) functions.

3.4 Examples of Newton's Method

Here are two numerical examples of Newton's method.

1. <u>Example #1</u>. A sample of data ($n = 20$) was generated from the intrinsically nonlinear regression model

$$y_t = \theta_1 + \theta_2 x_{2t} + \theta_2^2 x_{3t} + \epsilon_t,$$

where $\theta_1 = \theta_2 = 1$. The objective is to minimize the function

$$G(\theta_1, \theta_2) = (y - h(\theta_1, \theta_2))'(y - h(\theta_1, \theta_2))$$

where $h(\theta_1, \theta_2) = \theta_1 + \theta_2 x_{2t} + \theta_2^2 x_{3t}$. See Figure B.2 and Table B.3 (attached) to see how Newton's method performs for three different initial values.

2. Example #2. The objective is to minimize

$$G(\theta) = \theta^3 - 3\theta^2 + 5.$$

The gradient and Hessian are given by

$$
\begin{aligned}
g(\theta) &= 3\theta^2 - 6\theta = 3\theta(\theta - 2) \\
H(\theta) &= 6\theta - 6 = 6(\theta - 1).
\end{aligned}
$$

Substituting these into Newton's algorithm gives

$$\theta_{t+1} = \theta_t - \frac{3\theta_t(\theta_t - 2)}{6(\theta_t - 1)} = \theta_t - \frac{\theta_t(\theta_t - 2)}{2(\theta_t - 1)}.$$

Now consider two different starting values $\theta_0 = 1.5$ and $\theta_0 = 0.5$.

Starting Value $\theta_0 = 1.5$	Starting Value $\theta_0 = 0.5$
$\theta_1 = 1.5 - \frac{(1.5)(-0.5)}{2(0.5)} = 2.25$	$\theta_1 = 0.5 - \frac{(0.5)(-1.5)}{2(-0.5)} = -0.25$
$\theta_2 = 2.25 - \frac{(2.25)(0.25)}{2(1.25)} = 2.025$	$\theta_2 = -0.25 - \frac{(-0.25)(-2.25)}{2(-1.25)} = -0.025$
$\theta_3 = 2.025 - \frac{(2.025)(0.025)}{2(1.025)} = 2.0003$	$\theta_3 = -0.025 - \frac{(-0.025)(-2.025)}{2(-1.025)} = -0.0003.$

This examples highlights the fact that, at least for objective functions that are not globally concave (or convex), the choice of starting values is an important aspect of nonlinear optimization.

4   MATLAB Example

In this example, we are going to estimate the parameters of an intrinsically nonlinear Cobb-Douglas production function

$$Q_t = \beta_1 L_t^{\beta_2} K_t^{\beta_3} + \epsilon_t$$

using Gauss-Newton and Newton's method, as well as test for constant returns to scale (see <u>MATLAB example 14</u>).

For Gauss-Newton, the relevant gradient vector is

$$g(X, \beta^0) = \{L_t^{\beta_2^0} K_t^{\beta_3^0}, \ln(L_t)\beta_1^0 L_t^{\beta_2^0} K_t^{\beta_3^0}, \ln(K_t)\beta_1^0 L_t^{\beta_2^0} K_t^{\beta_3^0}\}.$$

For Newton's method, the relevant gradient and Hessian matrices are

$$
\begin{aligned}
g(b) &= -2\sum_{i=1}^{n} e_i\left(\frac{\partial h(x_i, b)}{\partial b}\right) \\
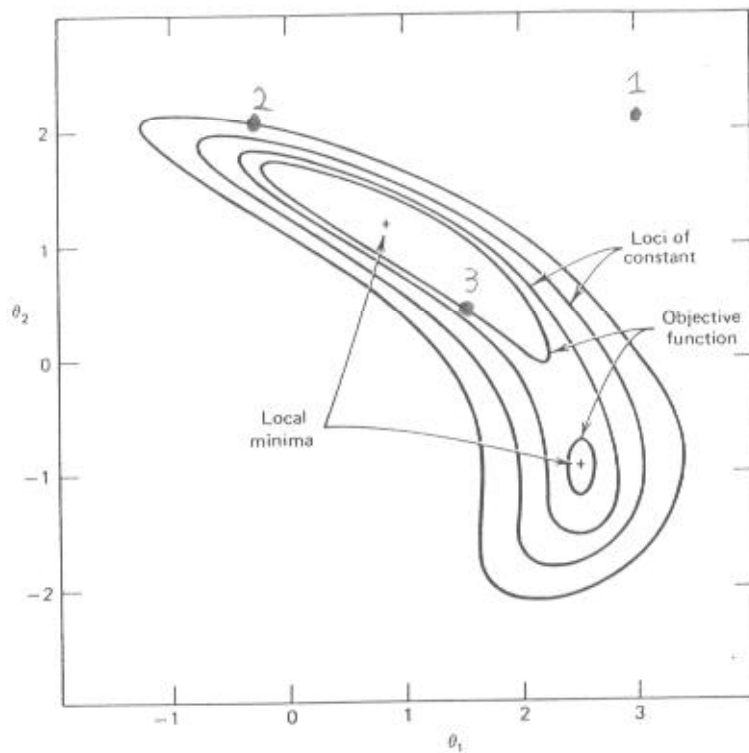H(b) &= \frac{\partial g(b)}{\partial b}.
\end{aligned}
$$

Figure B.2 Loci of constant objective function $H(\theta)$.

TABLE B.3 ITERATIONS OF THE NEWTON ALGORITHM

| n	$\theta_{n,1}$	$\theta_{n,2}$	$H(\theta_n)$
1	3.000000	2.000000	264.3918
2	-0.084033	1.811210	20.6328
3	0.625029	1.423940	16.5105
4	0.817259	1.272776	16.0961
5	0.862590	1.237516	16.0818
6	0.864782	1.235753	16.0817
7	0.864787	1.235748	16.0817
8	0.864787	1.235748	16.0817
1	0.000000	2.000000	29.2758
2	0.334936	1.600435	17.7382
3	0.735040	1.336953	16.1955
4	0.849677	1.247743	16.0832
5	0.864541	1.235946	16.0817
6	0.864787	1.235749	16.0817
7	0.864787	1.235748	16.0817
8	0.864787	1.235748	16.0817
1	1.500000	0.500000	20.2951
2	2.256853	0.007135	20.7735
3	2.467047	-0.436460	21.0312
4	2.316982	-0.202435	20.9467
5	2.359743	-0.320579	20.9809
6	2.354457	-0.319153	20.9805
7	2.354471	-0.319186	20.9805
8	2.354471	-0.319186	20.9805
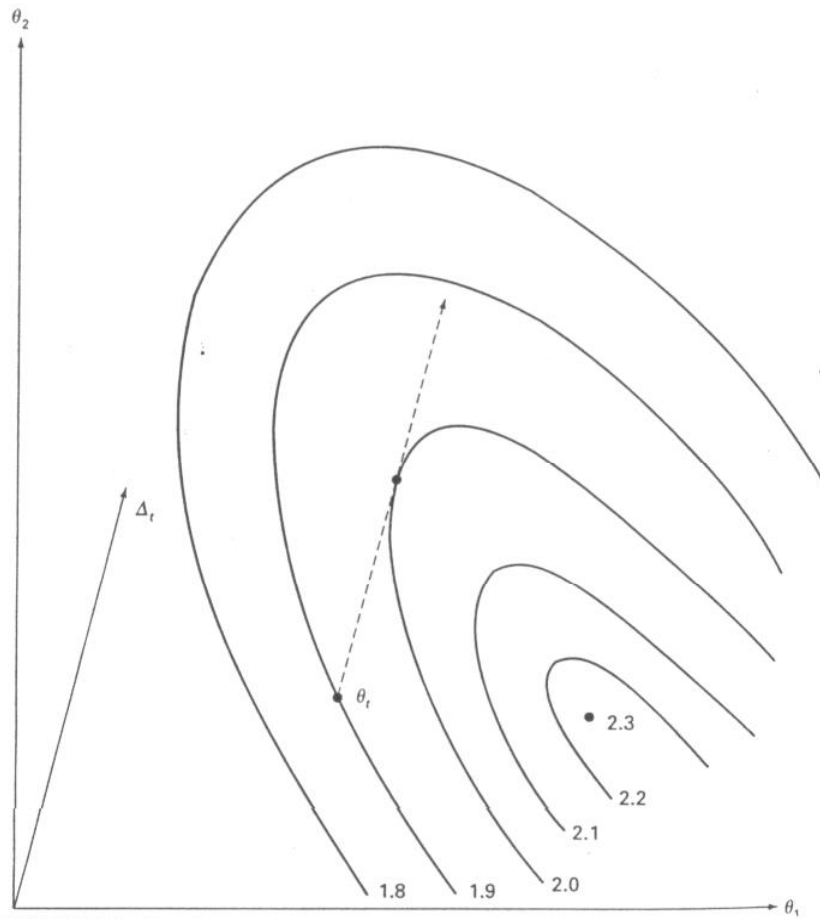
FIGURE 5.3   Iteration.

FIGURE 5.4   Line Search.