On-Line Supp.

How do we proceed to maximize the log-likelihood function of a sample by selecting the optimizing parameters, subject to $\boldsymbol{\theta} \in \Theta$? Appropriate **methods of numerical, constrained optimization** need to be implemented: this is what packages such as Matlab, Gauss , EViews, or Stata are for. For instance (i.e., other, better but more complex methods are feasible), **Newton's method** makes use of the **Hessian**, which is a $K \times K$ matrix $H(\boldsymbol{\theta}) \equiv \partial^2 \ell(\boldsymbol{\theta}) / \partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'$ that collects second partial derivatives of the log-likelihood function with respect to each of the parameters in $\boldsymbol{\theta}$. Similarly the $K \times 1$ **gradient** $\partial \ell(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}$ collects the first partial derivatives of the log-likelihood function with respect to each of the elements in $\boldsymbol{\theta}$. Let $\hat{\boldsymbol{\theta}}_j$ denote the value of the vector of estimates at step *j* of the algorithm, and let $\partial \ell(\hat{\boldsymbol{\theta}}_j) / \partial \boldsymbol{\theta}$ and $\mathrm{H}(\hat{\boldsymbol{\theta}}_j)$ denote, respectively, the gradient and the Hessian evaluated at $\hat{\boldsymbol{\theta}}_j$. Then the fundamental equation to update the estimates according to Newton's algorithm is:

$$\hat{\boldsymbol{\theta}}_{j+1} = \hat{\boldsymbol{\theta}}_j - \mathrm{H}^{-1}(\hat{\boldsymbol{\theta}}_j)[\partial \ell(\hat{\boldsymbol{\theta}}_j) / \partial \boldsymbol{\theta}] \qquad (0.1)$$

Because the log-likelihood function is to be maximized, the Hessian should be negative definite, at least when $\hat{\boldsymbol{\theta}}_j$ is sufficiently near $\hat{\boldsymbol{\theta}}_T$. This ensures that this step is in an uphill direction. The maximization process therefore proceeds through the following steps:

- Set an initial vector of parameters, $\hat{\boldsymbol{\theta}}_0$, and compute $\mathrm{H}^{-1}(\hat{\boldsymbol{\theta}}_0)$ and $\partial \ell(\hat{\boldsymbol{\theta}}_0) / \partial \boldsymbol{\theta}$.
- Compute the new vector of estimated parameters $\hat{\boldsymbol{\theta}}_1 = \hat{\boldsymbol{\theta}}_0 - \mathrm{H}^{-1}(\hat{\boldsymbol{\theta}}_0)[\partial \ell(\hat{\boldsymbol{\theta}}_0) / \partial \boldsymbol{\theta}]$ and therefore $\mathrm{H}^{-1}(\hat{\boldsymbol{\theta}}_1)$ and $\partial \ell(\hat{\boldsymbol{\theta}}_1) / \partial \boldsymbol{\theta}$; check that the **Euclidean norm** $|| \hat{\boldsymbol{\theta}}_1 - \hat{\boldsymbol{\theta}}_0 ||$ (in words, this is the square root of the sum of all squared differences between the elements of $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_0$) is not inferior to some small threshold parameter (typically, 10^{-5}).
- Update the vector of parameter estimates to $\hat{\boldsymbol{\theta}}_2 = \hat{\boldsymbol{\theta}}_1 - \mathrm{H}^{-1}(\hat{\boldsymbol{\theta}}_1)[\partial \ell(\hat{\boldsymbol{\theta}}_1) / \partial \boldsymbol{\theta}]$ and check that the norm $|| \hat{\boldsymbol{\theta}}_2 - \hat{\boldsymbol{\theta}}_1 ||$ is not inferior to the threshold parameter.
- Continue (unless a maximum number of iteration has been

exceeded, but with fast computers often thousands of iterations are affordable in the space of a few minutes only) until $\hat{\boldsymbol{\theta}}_j = \hat{\boldsymbol{\theta}}_{j-1} - \mathrm{H}^{-1}(\hat{\boldsymbol{\theta}}_{j-1})[\partial \ell(\hat{\boldsymbol{\theta}}_{j-1}) / \partial \boldsymbol{\theta}]$ is such that $|| \hat{\boldsymbol{\theta}}_j - \hat{\boldsymbol{\theta}}_{j-1} ||$ falls below the fixed convergence threshold, that signals that the optimizing vector has stopped changing.

- Set $\hat{\boldsymbol{\theta}}_T^{ML} = \hat{\boldsymbol{\theta}}_j$.

Numerical optimization is a very sensitive business; a myriad of choices are considered to be crucial to obtain "reliable" results, such as the initial value $\hat{\boldsymbol{\theta}}_0$, the convergence tolerance criterion, and often how much the algorithm is supposed to "travel" in the direction indicated by the inverse Hessian matrix, i.e., the coefficient τ in the iteration in (5.123), generalized to read as $\hat{\boldsymbol{\theta}}_{j+1} = \hat{\boldsymbol{\theta}}_j - \tau \mathrm{H}^{-1}(\hat{\boldsymbol{\theta}}_j)[\partial \ell(\hat{\boldsymbol{\theta}}_j) / \partial \boldsymbol{\theta}]$, where $\tau > 0$ (clearly a $\tau < 1$ "dims" the step taken in direction $[\partial \ell(\hat{\boldsymbol{\theta}}_j) / \partial \boldsymbol{\theta}]$, while a $\tau > 1$ acts as a multiplier). Reliability here is often evidence or even taken to offer some guarantee that $\hat{\boldsymbol{\theta}}_T^{ML} = \hat{\boldsymbol{\theta}}_j$ truly represents a **global** (as opposed to local) **maximizer** of the log-likelihood function and as such it is unique, as assumed. For instance, just to get hard evidence on this aspect, it is often advised to start off the maximization algorithm in correspondence of a range of alternative starting values and then retain, for the true and often lengthy iterative Netwon-style search, the most promising one(s).

Other numerical optimization methods are of course possible. A few of them are faster than Newton's method because they replace the Hessian matrix with cheaper to compute negative definite $K \times K$ matrices, for instance $OPG(\boldsymbol{\theta}) \equiv -[\partial \ell(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}][\partial \ell(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}]'$, which is negative definite by construction, unless $\partial \ell(\boldsymbol{\theta}) / \partial \boldsymbol{\theta} = \mathbf{0}$, which would instead show that a stationary point has been reached. The advantage of this expression is that it only requires calculation (often numerically) of first-order derivatives. Moreover, our simplified illustration of Newton's method ignores the role played by constraints, that may interfere with setting $\hat{\boldsymbol{\theta}}_j = \hat{\boldsymbol{\theta}}_{j-1} - \mathrm{H}^{-1}(\hat{\boldsymbol{\theta}}_{j-1})\ [\partial \ell(\hat{\boldsymbol{\theta}}_{j-1}) / \partial \boldsymbol{\theta}]$, when the constraints are violated.