Step-size

It is possible for the N-R procedure, as for other procedure below, to step past the maximum and move to a lower $LL(\beta)$. Figure 8.4 depicts the situation. The actual LL is given by the solid line. The dotted line is a quadratic function that has the slope and curvature that LL has at the point β_t. The N-R procedure moves to the top of the quadratic, to β_{t+1}. However, $LL(\beta_{t+1})$ is lower than $LL(\beta_t)$ in this case.

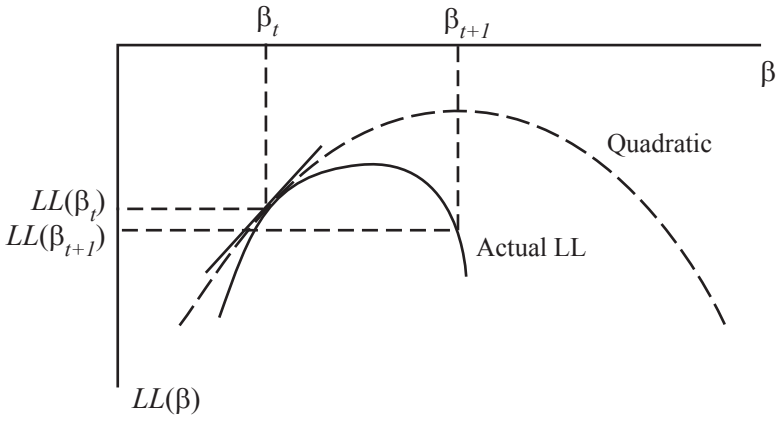

Figure 8.4: Step may go beyond maximum to lower LL.

To account for this possibility, the step is multiplied by a scalar λ in the N-R formula:

$$\beta_{t+1} = \beta_t + \lambda(-H_t)^{-1}g_t.$$

The vector $(-H_t)^{-1}g_t$ is called the direction and λ is called the step-size. (This terminology is standard even though $(-H_t)^{-1}g_t$ contains step-size information through H_t, as explained above in relation to Figure 8.3.) The step-size λ is reduced to assure that each step of the N-R procedure provides an increase in $LL(\beta)$. The adjustment is performed separately in each iteration, as follows.

Start with $\lambda = 1$. If $LL(\beta_{t+1}) > LL(\beta_t)$, move to β_{t+1} and start a new iteration. If $LL(\beta_{t+1}) < LL(\beta_t)$, then set $\lambda = 1/2$ and try again. If with $\lambda = 1/2$, $LL(\beta_{t+1})$ is still below $LL(\beta_t)$, then set $\lambda = 1/4$ and try again. Continue this process until a λ is found for which $LL(\beta_{t+1}) > LL(\beta_t)$. If this process results in a tiny λ, then little

progress is made in finding the maximum. This can be taken as a signal to the researcher that a different iteration procedure might be needed.

An analogous step-size adjustment can be made in the other direction, that is, by increasing λ when appropriate. A case is shown in Figure 8.5. The top of the quadratic is obtained with a step-size

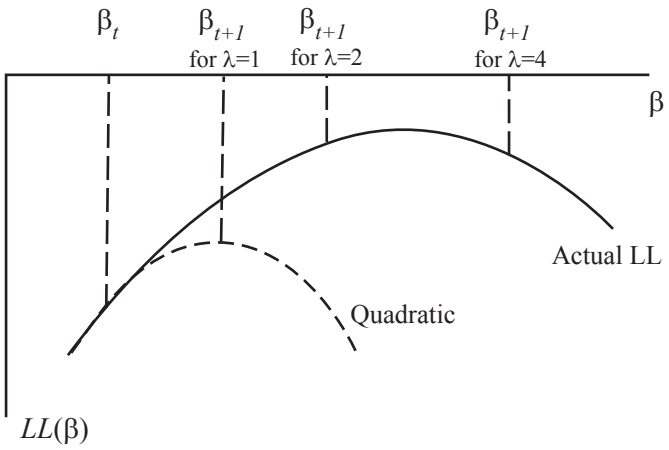

Figure 8.5: Double λ as long as LL rises.

of $\lambda = 1$. However, the $LL(\beta)$ is not quadratic and its maximum is further away. The step-size can be adjusted upward as long as the $LL(\beta)$ continues to rise. That is, calculate β_{t+1} with $\lambda = 1$ at β_{t+1}. If $LL(\beta_{t+1}) > LL(\beta_t)$, then try $\lambda = 2$. If the β_{t+1} based on $\lambda = 2$ gives a higher value of the log-likelihood function than with $\lambda = 1$, then try $\lambda = 4$. And so on, doubling λ as long as doing so further raises the likelihood function. Each time, $LL(\beta_{t+1})$ with a doubled λ is compared to its value at the previously tried λ, rather than with $\lambda = 1$, in order to assure that each doubling raises the likelihood function further than it had previously been raised with smaller λ's. In Figure 8.5, a final step-size of 2 is used, since the likelihood function with $\lambda = 4$ is lower than when $\lambda = 2$, even though it is higher than with $\lambda = 1$.

The advantage of this approach of raising λ is that it usually reduces the number of iterations that are needed to reach the maximum. New values of λ can be tried without re-calculating g_t and H_t, while each new iteration requires calculation of these terms. Adjusting λ can

therefore quicken the search for the maximum.

Concavity

If the log-likelihood function is globally concave, then the N-R procedure is guaranteed to provide an increase in the likelihood function at each iteration. This fact is demonstrated as follows. $LL(\beta)$ being concave means that its Hessian is negative definite at all values of β. (In one dimension, the slope of $LL(\beta)$ is declining such that the second derivative is negative.) If H is negative definite, then H^{-1} is also negative definite, and $(-H^{-1})$ is positive definite. By definition, a symmetric matrix M is positive definite if $x'Mx > 0$ for any $x \neq 0$. Consider a first-order Taylor's approximation of $LL(\beta_{t+1})$ around $LL(\beta_t)$:

$$LL(\beta_{t+1}) = LL(\beta_t) + (\beta_{t+1} - \beta_t)' g_t.$$

Under the N-R procedure, $\beta_{t+1} - \beta_t = \lambda(-H_t^{-1})g_t$. Substituting gives:

$$\begin{aligned} LL(\beta_{t+1}) &= LL(\beta_t) + (\lambda(-H_t^{-1})g_t)' g_t \\ &= LL(\beta_t) + \lambda g_t'(-H_t^{-1})g_t. \end{aligned}$$

Since $-H^{-1}$ is positive definite, the quantity $g_t'(-H_t^{-1})g_t > 0$ and $LL(\beta_{t+1}) > LL(\beta_t)$. Note that since this comparison is based on a first-order approximation, an increase in $LL(\beta)$ might only be obtained in a small neighborhood of β_t. That is, the value of λ that provides an increase might be small. However, an increase is indeed guaranteed at each iteration if $LL(\beta)$ is globally concave.

Suppose the log-likelihood function has regions that are not concave. In these areas, the N-R procedure can fail to find an increase. If the function is convex at β_t, then the N-R procedure moves in the opposite direction of the slope of the log-likelihood function. The situation is illustrated in Figure 8.6 for $K = 1$. The N-R step with one parameter is $LL'(\beta)/(-LL''(\beta))$, where the prime denotes derivatives. The second derivative is positive at β_t since the slope is rising. Therefore, $-LL''(\beta)$ is negative, and the step is in the opposite direction of the slope. With $K > 1$, if the Hessian is positive definite at β_t, then $(-H_t^{-1})$ is negative definite, and N-R steps in the opposite direction of g_t.

The sign of the Hessian can be reversed in these situations. However, there is no reason for using the Hessian where the function is not concave, since the Hessian in convex regions does not provide any

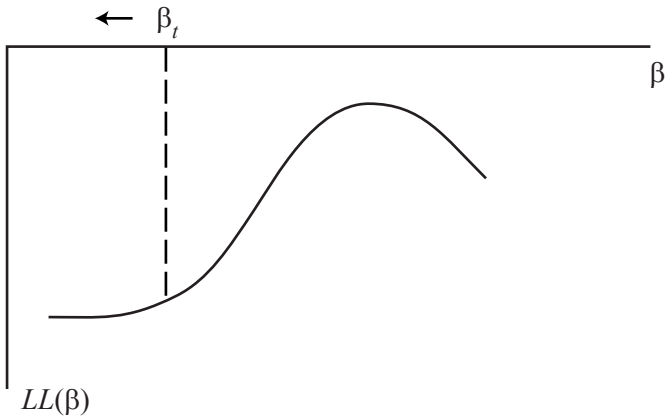

Figure 8.6: N-R in convex portion of LL.

useful information on where the maximum might be. There are easier ways to find an increase in these situations than calculating the Hessian and reversing its sign. This issue is part of the motivation for other procedures.

The N-R procedure has two drawbacks. First, calculation of the Hessian is usually computationally intensive. Procedures that avoid calculating the Hessian at every iteration can be much faster. Second, as we have just shown, the N-R procedure does not guarantee an increase in each step if the log-likelihood function is not globally concave. When $(-H_t^{-1})$ is not positive definite, an increase is not guaranteed.

Other approaches use approximations to the Hessian that address these two issues. The methods differ in the form of the approximation. Each procedure defines a step as:

$$\beta_{t+1} = \beta_t + \lambda M_t g_t,$$

where M_t is a $K \times K$ matrix. For N-R, $M_t = -H^{-1}$. Other procedures use M_t's that are easier to calculate than the Hessian and are necessarily positive definite so as to guarantee an increase at each iteration even in convex regions of the log-likelihood function.

8.3.2 BHHH

The N-R procedure does not utilize the fact that the function being maximized is actually the sum of log-likelihoods over a sample of observations. The gradient and Hessian are calculated just as one would do in maximizing any function. This characteristic of N-R provides generality, in that the N-R procedure can be used to maximize any function, not just a log-likelihood. However, as we will see, maximization can be faster if we utilize the fact that the function being maximized is a sum of terms in a sample.

We need some additional notation to reflect the fact that the log-likelihood function is a sum over observations. The score of an observation is the derivative of that observation's log likelihood with respect to the parameters: $s_n(\beta_t) = \partial lnP_n(\beta)/\partial\beta$ evaluated at β_t. The gradient, which we defined above and used for the N-R procedure, is the average score: $g_t = \sum_n s_n(\beta_t)/N$. The outer product of observation n's score is the $K \times K$ matrix:

$$s_n(\beta_t)s_n(\beta_t)' = \begin{pmatrix} s_n^1 s_n^1 & s_n^1 s_n^2 & \cdots & s_n^1 s_n^K \\ s_n^1 s_n^2 & s_n^2 s_n^2 & \cdots & s_n^2 s_n^K \\ \cdots & & & \\ s_n^1 s_n^K & s_n^2 s_n^K & \cdots & s_n^K s_n^K \end{pmatrix}$$

where s_n^k is the k-th element of $s_n(\beta_t)$ with the dependence on β_t omitted for convenience. The average outer product in the sample is $B_t = \sum_n s_n(\beta_t)s_n(\beta_t)'/N$. This average outer product is related to the covariance matrix: if the average score were zero, then B would be the covariance matrix of scores in the sample. Often B_t is called the "outer product of the gradient." This term can be confusing since B_t is not the outer product of g_t. However, the term reflects the fact that the score is an observation-specific gradient and B_t is the average outer product of these observation-specific gradients.

At the parameters that maximize of the likelihood function, the average score is indeed zero. The maximum occurs where the slope is zero, which means that the gradient, i.e., the average score, is zero. Since the average score is zero, the outer product of the scores, B_t, becomes the variance of the scores. That is, at the maximizing values of the parameters, B_t is the variance of scores in the sample.

The variance of the scores provides important information for locating the maximum of the likelihood function. In particular, this variance provides a measure of the curvature of the log-likelihood function,

similar to the Hessian. Suppose that everyone in the sample has similar scores. Since everyone is fairly similar, the sample contains very little information. The log-likelihood function is fairly flat in this situation, reflecting the fact that the sample contains little information such that different values of the parameters fit the data about the same. The first panel of Figure 8.7 depicts this situation: with a fairly flat log-likelihood, different values of β give similar values of $LL(\beta)$. The

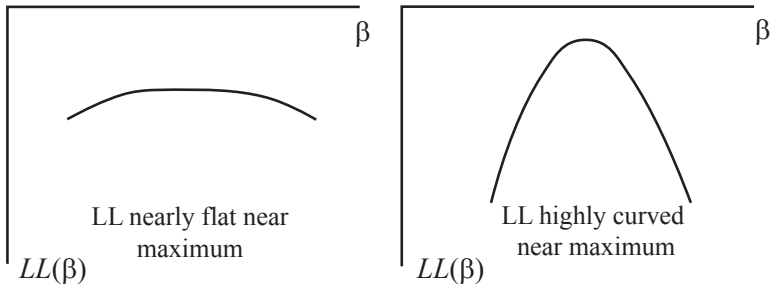

Figure 8.7: Shape of log-likelihood function near maximum.

curvature is small when the variance of the scores is small. Conversely, the scores differing greatly over observations means that the observations are quite different and the sample provides considerable amount of information. The log-likelihood function is highly peaked, reflecting the fact that the sample provides good information on the values of β. Moving away from the maximizing values of β causes a large loss of fit. The second panel of Figure 8.7 illustrates this situation. The curvature is great when the variance of the scores is high.

These ideas about the variance of the scores and their relation to the curvature of the log-likelihood function are formalized in the famous "information identity." This identity states that the covariance of the scores at the true parameters is equal to the negative of the expected Hessian. We demonstrate this identity in the last section of this chapter; Theil (1971) and Ruud (2000) also provide useful and heuristic proofs. However, even without proof, it makes intuitive sense that the variance of the scores provides information on the curvature of the log-likelihood function.

Berndt, Hall, Hall and Hausman (1974), hereafter referred to as BHHH (and commonly pronounced B-triple H), proposed using this relationship in the numerical search for the maximum of the log-likelihood

function. In particular, the BHHH procedure uses B_t in the optimization routine in place of $-H_t$. Each iteration is defined by:

$$\beta_{t+1} = \beta_t + \lambda B_t^{-1} g_t.$$

This step is the same as for N-R except that B_t is used in place of $-H_t$. Given the discussion above about the variance of the scores indicating the curvature of the log-likelihood function, replacing $-H_t$ with B_t makes sense.

There are two advantages to the BHHH procedure over N-R:

(1) B_t is far faster to calculate that H_t. The scores must be calculated to obtain the gradient for the N-R procedure anyway, and so calculating B_t as the average outer product of the scores takes hardly any extra computer time. In contrast, calculating H_t requires calculating the second derivatives of the log-likelihood function.

(2) B is necessarily positive definite. The BHHH procedure is therefore guaranteed to provide an increase in $LL(\beta)$ in each iteration, even in convex portions of the function. Using the proof given above for N-R when $-H_t$ is positive definite, the BHHH step $\lambda B_t^{-1} g_t$ raises $LL(\beta)$ for a small enough λ.

Our discussion about the relation of the variance of the scores to the curvature of the log-likelihood function can be stated a bit more precisely. For a correctly specified model at the true parameters, $B \to -H$ as $N \to \infty$. This relation between the two matrices is an implication of the information identity, discussed at greater length in the last section. This convergence suggests that B_t can be considered an approximation to $-H_t$. The approximation is expected to be better as sample size rises. And the approximation can be expected to be better close to the true parameters, where the expected score is zero and the information identity holds, than for values of β that are farther from their true values. That is, B_t can be expected to be a better approximation close to the maximum of the $LL(\beta)$ than farther from the maximum.

There are some drawbacks of BHHH. The procedure can give small steps that raise $LL(\beta)$ very little, especially when the iterative process is far from the maximum. This behavior can arise because B_t is not a good approximation to $-H_t$ far from the true value. Or it might arise because $LL(\beta)$ is highly non-quadratic in the area where the problem is occurring. If the function is highly non-quadratic, N-R does not perform well as explained above; since BHHH is an approximation to N-R,

BHHH would not perform well even if B_t were a good approximation to $-H_t$.

8.3.3 BHHH-2

The BHHH procedure relies on the matrix B_t which, as we have described, captures the covariance of the scores when the average score is zero (i.e., at the maximizing value of β.) When the iterative process is not at the maximum, the average scores is not zero and B_t does not represent the covariance of the scores.

A variant on the BHHH procedure is obtained by subtracting out the mean score before taking the outer product. For any level of the average score, the covariance of the scores over the sampled decision-makers is

$$W_t = \sum_n (s_n(\beta_t) - g_t)(s_n(\beta_t) - g_t)'/N$$

where the gradient g_t is the average score. W_t is the covariance of the scores around their mean, and B_t is the average outer product of the scores. W_t and B_t are the same when the mean gradient is zero (i.e., at the maximizing value of β), but differ otherwise.

The maximization procedure can use W_t instead of B_t:

$$\beta_{t+1} = \beta_t + \lambda W_t^{-1} g_t.$$

This procedure, which I call BHHH-2, has the same advantages as BHHH. W_t is necessarily positive definite, since it proportional to a covariance matrix, and so the procedure is guaranteed to provide an increase in $LL(\beta)$ at every iteration. Also, for a correctly specified model at the true parameters, $W \to -H$ as $N \to \infty$, such that W_t can be considered an approximation to $-H_t$. The information identity establishes this equivalence, as for B.

For β's that are close to the maximizing value, BHHH and BHHH-2 give nearly the same results. They can differ greatly at values far from the maximum. Experience indicates, however, that the two methods are fairly similar in that either both of them work effectively for a given likelihood function, or neither of them does. The main value of BHHH-2 is pedagogical, to elucidate the relation between the covariance of the scores and the average outer product of the scores. This relation is critical in the analysis of the information identity in the last section.

8.3.4 Steepest Ascent

This procedure is defined by the iteration formula:

$$\beta_{t+1} = \beta_t + \lambda g_t.$$

The defining matrix for this procedure is the identity matrix I. Since I is positive definite, the method guarantees an increase in each iteration. It is called "steepest ascent" because it provides the greatest possible increase in $LL(\beta)$ for the distance between β_t and β_{t+1}, at least for small enough distance. Any other step of the same distance provides less increase. This fact is demonstrated as follows. Take a first-order Taylor's expansion of $LL(\beta_{t+1})$ around $LL(\beta_t)$: $L(\beta_{t+1}) = \beta_t + (\beta_{t+1} - \beta_t)g_t$. Maximize this expression for $LL(\beta_{t+1})$ subject to the Euclidian distance from β_t to β_{t+1} being $\sqrt{k}$. That is, maximize subject to $(\beta_{t+1} - \beta_t)'(\beta_{t+1} - \beta_t) = k$. The Lagrangian is

$$\mathrm{L} = LL(\beta_t) + (\beta_{t+1} - \beta_t)g_t - \frac{1}{2\lambda}[(\beta_{t+1} - \beta_t)'(\beta_{t+1} - \beta_t) - k]$$

$$\begin{aligned} \frac{\partial \mathrm{L}}{\partial \beta_{t+1}} &= g_t - \frac{1}{\lambda}(\beta_{t+1} - \beta_t) = 0 \\ \beta_{t+1} - \beta_t &= \lambda g_t \\ \beta_{t+1} &= \beta_t + \lambda g_t \end{aligned}$$

which is the formula for steepest ascent.

While at first encounter, one might think that the method of steepest ascent is the best possible procedure since it gives the greatest possible increase in the log-likelihood function at each step. However, the method's property is actually less grand than this statement implies. Note that the derivation relies on a first-order approximation that is only accurate in a neighborhood of β_t. The correct statement of the result is that there is some sufficiently small distance for which the method of steepest ascent gives the greatest increase for that distance. This distinction is critical. Experience indicates that the step-sizes are often very small with this method. The fact that the ascent is greater than for any other step of the same distance is not particularly comforting when the steps are so small. Usually, BHHH and BHHH-2 converge more quickly than the method of steepest ascent.

8.3.5 DFP and BFGS

The methods of Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) calculate the approximate Hessian is a way that uses information at more than one point on the likelihood function. Recall that N-R uses the actual Hessian at β_t to determine the step to β_{t+1}, and BHHH and BHHH-2 use the gradient matrix at β_t to approximate the Hessian. Only information at β_t is being used to determine the step in these procedures. If the function is quadratic, then information at one point on the function provides all the information that is needed about the shape of the function. These methods work well, therefore, when the log-likelihood function is close to quadratic. In contrast, the DFP and BFGS procedures use information at several points to obtain a sense of the curvature of the log-likelihood function.

The Hessian is the matrix of second derivatives. As such, it gives the amount by which the slope of the curve changes as one moves along the curve. The Hessian is defined for infinitesimally small movements. Since we are interested in making large steps, understanding how the slope changes for non-infinitesimal movements is useful. An "arc" Hessian can be defined on the basis of how the gradient changes from one point to another. For example, for function $f(x)$, suppose the slope at $x = 3$ is 25 and at $x = 4$ the slope is 19. The change in slope for a one unit change in x is -6. In this case, the arc Hessian is -6, representing the change in the slope as a step is taken from $x = 3$ to $x = 4$.

The procedures of DFP and BFGS use these concepts to approximate the Hessian. The gradient is calculated at each step in the iteration process. The difference in the gradient between the various points that have been reached is used to calculate an arc Hessian over these points. This arc Hessian reflects the actual change in gradient that occurs for actual movement on the curve, as opposed to the Hessian which simply reflects the change in slope for infinitesimally small steps around that point. When the log-likelihood function is non-quadratic, the Hessian at any point provides little information about the shape of the function. The arc Hessian provides better information.

At each iteration, the DFP and BFGS procedures update the arc Hessian using information that is obtained at the new point, that is, using the new gradient. The two procedures differ in how the updating in performed; see Greene (2000) for details. Both methods are extremely effective, usually far more efficient that N-R, BHHH, BHHH-

2, or steepest ascent. BFGS refines DFP, and my experience indicates that it nearly always works better. BFGS is the default algorithm in the optimization routines of many commercial software packages.

8.4 Convergence criterion

In theory the maximum of $LL(\beta)$ occurs when the gradient vector is zero. In practice, the calculated gradient vector is never exactly zero: it can be very close, but a series of calculations on a computer cannot produce a result of exactly zero (unless of course, the result is set to zero through a Boolean operator or by multiplication by zero, neither of which arises in calculation of the gradient.) The question arises: when are we sufficiently close to the maximum to justify stopping the iterative process?

The statistic $m_t = g_t'(-H_t^{-1})g_t$ is often used to evaluate convergence. The researcher specifies a small value for m, such as $\breve{m} = .0001$, and determines in each iteration whether $g_t'(-H_t^{-1})g_t < \breve{m}$. If this inequality is satisfied, the iterative process stops and the parameters at that iteration are considered the converged values, that is, the estimates. For procedures other than N-R that use an approximate Hessian in the iterative process, the approximation is used in the convergence statistic to avoid calculating the actual Hessian. Close to the maximum, where the criterion becomes relevant, each form of approximate Hessian that we have discussed is expected to be similar to the actual Hessian.

The statistic m_t is the test statistic for the hypothesis that all elements of the gradient vector are zero. The statistic is distributed chi-squared with K degrees of freedom. However, the convergence criterion $\breve{m}$ is usually set far more stringently (that is, lower) than the critical value of a chi-squared at standard levels of significance, so as to assure that the estimated parameters are very close to the maximizing values. Usually, the hypothesis that the gradient elements are zero cannot be rejected for a relatively wide area around the maximum. The distinction can be illustrated for an estimated coefficient that has a t-statistic of 1.96. The hypothesis cannot be rejected that this coefficient is any value between zero and twice its estimated value. However, we would not want convergence to be defined as having reached any parameter value within this range.

It is tempting to view small changes in β_t from one iteration to the

next, and correspondingly small increases in $LL(\beta_t)$, as evidence that convergence has been achieved. However, as stated above, the iterative procedures may produce small steps because the likelihood function is not close to a quadratic rather than because of nearing the maximum. Small changes in β_t and $LL(\beta_t)$ accompanied by a gradient vector that is not close to zero indicates that the numerical routine is not effective at finding the maximum.

Convergence is sometimes assessed on the basis of the gradient vector itself rather than through the test statistic m_t. There are two procedures: (1) determine whether each element of the gradient vector is smaller in magnitude than some value that the researcher specifies, and (2) divide each element of the gradient vector by the corresponding element of β and determine whether each of these quotients is smaller in magnitude than some value specified by the researcher. The second approach normalizes for the units of the parameters, which are determined by the units of the variables that enter the model.

8.5 Local versus global maximum

All of the methods that we have discussed are susceptible to converging at a local maximum that is not the global maximum, as shown in Figure 8.8. When the log-likelihood function is globally concave, as for logit with linear-in-parameters utility, then there is only one maximum and the issue doesn't arise. However, most discrete choice models are not globally concave.

A way to investigate the issue is to use a variety of starting values and observe whether convergence occurs at the same parameter values. For example, in Figure 8.8, starting at β_0 will lead to convergence at β_1. Unless other starting values were tried, the researcher would mistakenly believe that the maximum of $LL(\beta)$ had been achieved. Starting at β_2, convergence is achieved at $\hat{\beta}$. By comparing the $LL(\hat{\beta})$ with $LL(\beta_1)$, the researcher finds that β_1 is not the maximizing value. Liu and Mahmassani (2000) propose a way to select starting values that involves the researcher setting upper and lower bounds on each parameter and randomly choosing values within those bounds.

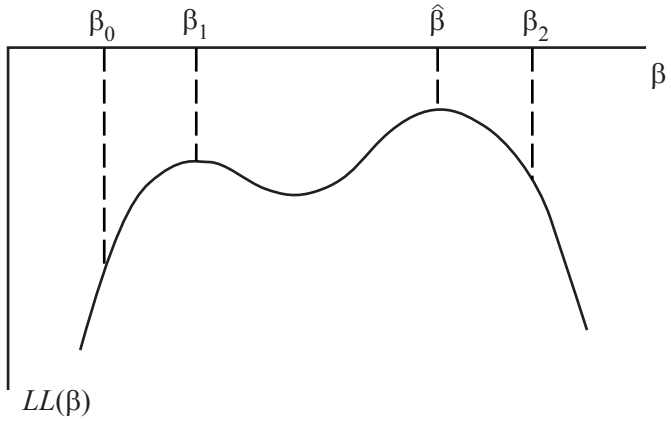

Figure 8.8: Local versus global maximum

8.6 Variance of the Estimates

In standard econometric courses, it is shown that, for a correctly specified model,

$$\sqrt{N}(\hat{\beta} - \beta^*) \xrightarrow{d} \mathrm{N}(0, (-\mathbf{H})^{-1})$$

as $N \to \infty$, where β^* is the true parameter vector, $\hat{\beta}$ is the maximum likelihood estimator, and $\mathbf{H}$ is the expected Hessian in the population. The negative of the expected Hessian, $-\mathbf{H}$, is often called the information matrix. Stated in words: the sampling distribution of the difference between the estimator and the true value, normalized for sample size, converges asymptotically to a normal distribution centered on zero and with covariance equal to the inverse of the information matrix, $-\mathbf{H}^{-1}$. Since the asymptotic covariance of $\sqrt{N}(\hat{\beta} - \beta^*)$ is $-\mathbf{H}^{-1}$, the asymptotic covariance of $\hat{\beta}$ itself is $-\mathbf{H}^{-1}/N$.

The bold-face type in these expressions indicates that $\mathbf{H}$ is the average in the population, as opposed to H which is the average Hessian in the sample. The researcher calculates the asymptotic covariance by using H as an estimate of $\mathbf{H}$. That is, the asymptotic covariance of $\hat{\beta}$ is calculated as $-H^{-1}/N$ where H is evaluated at $\hat{\beta}$.

Recall that W is the covariance of the scores in the sample. At the maximizing values of β, B is also the covariance of the scores. By the information identity discussed above and explained in the last section, $-H$, which is the (negative of the) average Hessian in the sample,

converges to the covariance of the scores for a correctly specified model at the true parameters. In calculating the asymptotic covariance of the estimates $\hat{\beta}$, any of these three matrices can be used as an estimate of $-\mathbf{H}$. The asymptotic variance of $\hat{\beta}$ is calculated as W^{-1}/N, B^{-1}/N, or $-H^{-1}/N$, where each of these matrices is evaluated at $\hat{\beta}$.

If the model is not correctly specified, then the asymptotic covariance of $\hat{\beta}$ is more complicated. In particular, for any model for which the expected score is zero at the true parameters,

$$\sqrt{N}(\hat{\beta} - \beta^*) \xrightarrow{d} \mathrm{N}(0, \mathbf{H}^{-1}\mathbf{V}\mathbf{H}^{-1})$$

where $\mathbf{V}$ is the variance of the scores in the population. When the model is correctly specified, the matrix $-\mathbf{H} = \mathbf{V}$ by the information identity, such that $\mathbf{H}^{-1}\mathbf{V}\mathbf{H}^{-1} = -\mathbf{H}^{-1}$ and we get the formula for a correctly specified model. However, if the model is not correctly specified, this simplification does not occur. The asymptotic distribution of $\hat{\beta}$ is $\mathbf{H}^{-1}\mathbf{V}\mathbf{H}^{-1}/N$. This matrix is called the "robust covariance matrix" since it is valid whether or not the model is correctly specified.

To estimate the robust covariance matrix, the researcher must calculate the Hessian H. If a procedure other than N-R is being used to reach convergence, the Hessian need not be calculated at each iteration; however, it must be calculated at the final iteration. Then the asymptotic covariance is calculated as $H^{-1}WH^{-1}$, or with B instead of W. This formula is sometimes called the "sandwich" estimator of the covariance, since the Hessian-inverse appears on both sides.

8.7 Information Identity

The information identity states that, for a correctly specified model at the true parameters, $\mathbf{V} = -\mathbf{H}$ where $\mathbf{V}$ is the covariance of the scores in the population and $\mathbf{H}$ is the average Hessian in the population. The score for a person is the vector of first derivatives of the the person's $lnP(\beta)$ with respect to the parameters, and the Hessian is the matrix of second derivatives. The information identity states that, in the population, the covariance of the first derivatives equals the average second derivatives (actually, the negative of these second derivatives.) This is a startling fact, not something that would be expected or even believed if there were not proof. It has implications throughout econometrics. The implications that we have used in the previous sections of this chapter are easily derivable from the identity. In particular:

(1) *At the maximizing value of β, $W \to -H$ as $N \to \infty$, where W is the sample covariance of the scores and H is the sample average of each observation's Hessian.* As sample size rises, the sample covariance approaches the population covariance: $W \to \mathbf{V}$. Similarly, the sample average of the Hessian approaches the population average: $H \to \mathbf{H}$. Since $\mathbf{V} = -\mathbf{H}$ by the information identity, W approaches the same matrix that $-H$ approaches, and so they approach each other.

(2) *At the maximizing value of β, $B \to -H$ as $N \to \infty$, where B is the sample average of the outer product of the scores.* At $\hat{\beta}$, the average score in the sample is zero, such that B is the same as W. The result for W applies for B.

We now demonstrate the information identity. We need to expand our notation to account for the population instead of simply the sample. Let $P_i(x, \beta)$ be the probability that a person who faces explanatory variables x chooses alternative i given the parameters β. Of the people in the population who face variables x, the share who choose alternative i is this probability calculated at the true parameters: $S_i(x) = P_i(x, \beta^*)$ where β^* are the true parameters. Consider now the gradient of $lnP_i(x, \beta)$ with respect to β. The average gradient in the population is

$$\mathbf{g} = \int \sum_i \frac{\partial lnP_i(x, \beta)}{\partial \beta} S_i(x) f(x) dx \tag{8.2}$$

where $f(x)$ is the density of explanatory variables in the population. This expression can be explained as follows. The gradient for people who face x and choose i is $\frac{\partial lnP_{ni}(\beta)}{\partial \beta}$. The average gradient is the average of this term over all values of x and all alternatives i. The share of people who face a given value of x is given by $f(x)$, and the share of people who face this x that choose i is $S_i(x)$. So, $S_i(x)f(x)$ is the share of the population who face x and choose i and therefore have gradient $\frac{\partial lnP_i(x,\beta)}{\partial \beta}$. Summing this term over all values of i and integrating over all values of x (assuming the x's are continuous) gives the average gradient, as expressed in (8.2).

The average gradient in the population is equal to zero at the true parameters. This fact can be considered either the definition of the true parameters or the result of a correctly specified model. Also, we know that $S_i(x) = P_i(x, \beta^*)$. Substituting these facts into (8.2), we

have:

$$0 = \int \sum_i \frac{\partial ln P_i(x,\beta)}{\partial \beta} P_i(x,\beta) f(x) dx,$$

where all terms are evaluated at β^*. We now take the derivative of this equation with respect to the parameters:

$$0 = \int \sum_i \left(\frac{\partial^2 ln P_i(x,\beta)}{\partial \beta \partial \beta'} P_i(x,\beta) + \frac{\partial ln P_i(x,\beta)}{\partial \beta} \frac{\partial P_i(x,\beta)}{\partial \beta'} \right) f(x) dx.$$

Since $\partial lnP/\partial\beta = (1/P)\partial P/\partial\beta$ by the rules of derivatives, we can substitute $\partial \frac{lnP_i(x,\beta)}{\partial \beta'} P_i(x,\beta)$ for $\frac{\partial P_i(x,\beta)}{\partial \beta'}$ in the last term in parentheses:

$$0 = \int \sum_i \left(\frac{\partial^2 ln P_i(x,\beta)}{\partial \beta \partial \beta'} P_i(x,\beta) + \frac{\partial ln P_i(x,\beta)}{\partial \beta} \frac{\partial ln P_i(x,\beta)}{\partial \beta'} P_i(x,\beta) \right) f(x) dx.$$

Rearranging,

$$-\int \sum_i \frac{\partial^2 ln P_i(x,\beta)}{\partial \beta \partial \beta'} P_i(x,\beta) f(x) dx = \int \sum_i \frac{\partial ln P_i(x,\beta)}{\partial \beta} \frac{\partial ln P_i(x,\beta)}{\partial \beta'} P_i(x,\beta) f(x) dx.$$

Since all terms are evaluated at the true parameters, we can replace $P_i(x,\beta)$ with $S_i(x)$ to obtain:

$$-\int \sum_i \frac{\partial^2 ln P_i(x,\beta)}{\partial \beta \partial \beta'} S_i(x) f(x) dx = \int \sum_i \frac{\partial ln P_i(x,\beta)}{\partial \beta} \frac{\partial ln P_i(x,\beta)}{\partial \beta'} S_i(x) f(x) dx.$$

The left hand side is the negative of the average Hessian in the population, $-\mathbf{H}$. The right hand side is the average outer product of the gradient, which is the covariance of the gradient since the average gradient is zero: $\mathbf{V}$. Therefore, $-\mathbf{H} = \mathbf{V}$, the information identity. As stated above, the matrix $-\mathbf{H}$ is often called the information matrix.

Chapter 9

Drawing from Densities

9.1 Introduction

Simulation consists of drawing from a density, calculating a statistic for each draw, and averaging the results. In all cases, the researcher wants to calculate an average of the form $\bar{t} = \int t(\varepsilon) f(\varepsilon)\, d\varepsilon$, where $t(\cdot)$ is a statistic of interest and $f(\cdot)$ is a density. To approximate this average through simulation, the researcher must be able to take draws from the density $f(\cdot)$. For some densities, this task is simple. However, in many situations, it is not immediately clear how to draw from the relevant density. Furthermore, even with simple densities, there might be ways of taking draws that provide a better approximation to the integral than a sequence of purely random draws.

We explore these issues in this chapter. In the first sections, we describe the most prominent methods that have been developed for taking purely random draws from various kinds of densities. These methods are presented in a progressive sequence, starting with simple procedures that work with a few convenient densities and moving to ever more complex methods that work with less convenient densities. The discussion culminates with the Metropolis-Hastings algorithm, which can be used with (practically) any density. The chapter then turns to the question of whether and how a sequence of draws can be taken that provides a better approximation to the relevant integral than a purely random sequence. We discuss antithetics, systematic sampling, and Halton sequences and show the value that these types of draws provide in estimation of model parameters.

9.2 Random Draws

9.2.1 Standard normal and uniform

If the researcher wants to take a draw from a standard normal density (that is, a normal with zero mean and unit variance) or a standard uniform density (uniform between 0 and 1), the process from a programming perspective is very easy. Most statistical packages contain random number generators for these densities. The researcher simply calls these routines to obtain a sequence of random draws. In the sections below, we refer to a draw of a standard normal as η and a draw of a standard uniform as μ.

The draws from these routines are actually called "pseudo-random numbers" because nothing that a computer does is truly random. There are many issues involved in the design of these routines. The intent in their design is to produce numbers that exhibit the properties of random draws. The extent to which this intent is realized depends, of course, on how one defines the properties of "random" draws. These properties are difficult to define precisely since randomness is a theoretical concept that has no operational counterpart in the real world. From a practical perspective, my advice is the following: unless one is willing to spend considerable time investigating and resolving (literally, re-solving) these issues, it is probably better to use the available routines rather than write a new one.

9.2.2 Transformations of standard normal

Some random variables are transformations of a standard normal. For example, a draw from a normal density with mean b and variance s^2 is obtained as $\varepsilon = b + s\eta$. A draw from a lognormal density is obtained by exponentiating a draw from a normal density: $\varepsilon = e^{(b+s\eta)}$. The moments of the lognormal are functions of the mean and variance of the normal that is exponentiated. In particular, the mean of ε is $\exp(b + (s^2/2))$ and its variance is $\exp(2b + s^2) \cdot (\exp(s^2) - 1)$. Given values for the mean and variance of the lognormal, the appropriate values of b and s to use in the transformation can be calculated. It is more common, however, to treat b and s as the parameters of the lognormal and calculate its mean and variance from these parameters.

9.2.3 Inverse cumulative for univariate densities

Consider a random variable with density $f(\varepsilon)$ and corresponding cumulative distribution $F(\varepsilon)$. If F is invertible (that is, if F^{-1} can be calculated), then draws of ε can be obtained from draws of a standard uniform. By definition, $F(\varepsilon) = k$ means that the probability of obtaining a draw equal to or below ε is k, where k is between zero and one. A draw μ from the standard uniform provides a number between zero and one. We can set $F(\varepsilon) = \mu$ and solve for the corresponding ε: $\varepsilon = F^{-1}(\mu)$. When ε is drawn in this way, the cumulative distribution of the draws is equal to F, such that the draws are equivalent to draws directly from F. An illustration is provided in Figure 9.1. A draw μ^1 from a standard uniform translates into the value of ε labeled ε^1, at which $F(\varepsilon^1) = \mu^1$.

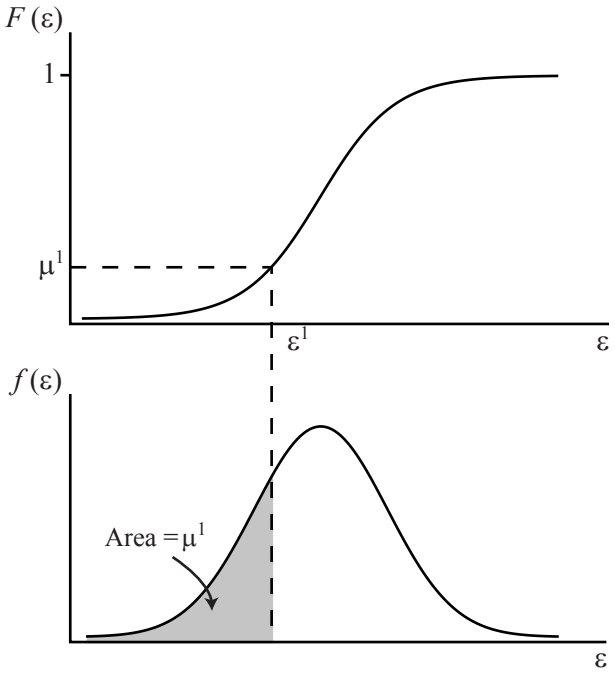

Figure 9.1: Draw of μ^1 from uniform and create $\varepsilon^1 = F^{-1}(\mu)$.

The extreme value distribution, which is the basis for multinomial logit models, provides an example. The density is $f(\varepsilon) = \exp(-\varepsilon) \cdot$

$\exp(-\exp(-\varepsilon))$ with cumulative distribution $F(\varepsilon) = \exp(-\exp(-\varepsilon))$. A draw from this density is obtained as $\varepsilon = -\ln(-\ln(\mu))$.

Note that this procedure works only for univariate distributions. If there are two or more elements of ε, then $F^{-1}(\mu)$ is not unique, since various combinations of the elements of ε have the same cumulative probability.

9.2.4 Truncated univariate densities

Consider a random variable that ranges from a to b with density proportional to $f(\varepsilon)$ within this range. That is, the denisty is $(1/k)f(\varepsilon)$ for $a \leq \varepsilon \leq b$, and 0 otherwise, where k is the normalizing constant that insures that the density integrates to 1: $k = \int_a^b g(\varepsilon)\, d\varepsilon = F(b) - F(a)$. A draw from this density can be obtained by applying the procedure in section 9.2.3 while assuring that the draw is within the appropriate range.

Draw μ from a standard uniform density. Calculate the weighted average of $F(a)$ and $F(b)$ as $\bar{\mu} = (1-\mu)F(a) + \mu F(b)$. Then calculate $\varepsilon = F^{-1}(\bar{\mu})$. Since $\bar{\mu}$ is between $F(a)$ and $F(b)$, ε is necessarily between a and b. Essentially, the draw of μ determines how far to go between a and b. Note that the normalizing constant k is not used in the calculations and therefore need not be calculated. Figure 9.2 illustrates the process.

9.2.5 Choleski transformation for multivariate normals

As described in section 9.2.2, a univariate normal with mean b and variance s^2 is obtained as $\varepsilon = b + s\mu$ where μ is standard normal. An analogous procedure can be used to draw from a multivariate normal. Let ε be a vector with K elements distributed $N(b, \Omega)$. A Choleski factor of Ω is defined as a lower-triangular matrix L such that $LL' = \Omega$. It is often called the generalized square root of Ω or generalized standard deviation of ε. With $K = 1$ and variance s^2, the Choleski factor is s, which is just the standard deviation of ε. Most statistical and matrix manipulation packages have routines to calculate a Choleski factor for any positive definite, symmetric matrix.

A draw of ε from $N(b, \Omega)$ is obtained as follows. Take K draws from a standard normal, and label the vector of these draws $\eta = \langle \eta_1, \ldots, \eta_K \rangle'$. Calculate $\varepsilon = b + L\eta$. We can verify the properties of ε. It is normally distributed since the sum of normals is normal.

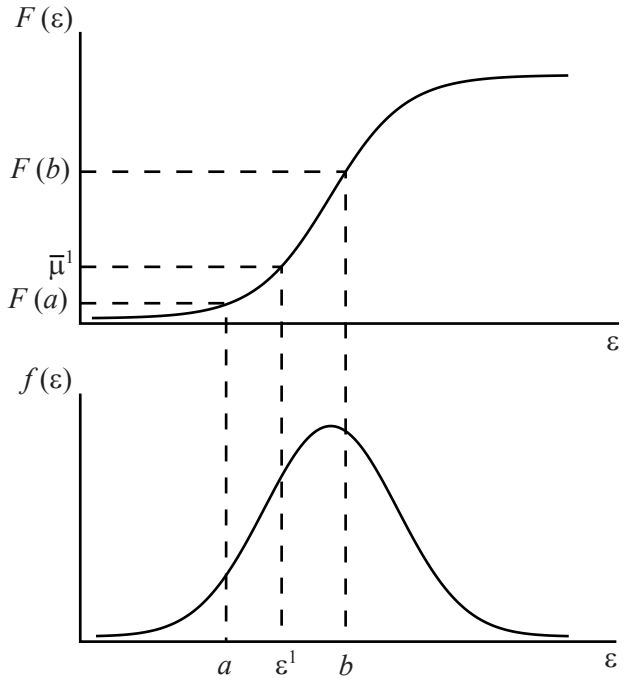

Figure 9.2: Draw of $\bar{\mu}^1$ between $F(a)$ and $F(b)$ gives draw ε^1 from $f(\varepsilon)$ between a and b.

Its mean is b: $\mathrm{E}(\varepsilon) = b + L\mathrm{E}(\eta) = b$. And its covariance is Ω: $\mathrm{Var}(\varepsilon) = \mathrm{E}(L\eta(\eta L)') = L\mathrm{E}(\eta\eta')L' = L\mathrm{Var}(\eta)L' = LIL' = LL' = \Omega$.

To be concrete, consider a three dimensional ε with zero mean. A draw of ε is calculated as

$$\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix} = \begin{pmatrix} s_{11} & 0 & 0 \\ s_{21} & s_{22} & 0 \\ s_{31} & s_{32} & s_{33} \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{pmatrix}$$

or

$$\begin{aligned} \varepsilon_1 &= s_{11}\eta_1 \\ \varepsilon_2 &= s_{21}\eta_1 + s_{22}\eta_2 \\ \varepsilon_3 &= s_{31}\eta_1 + s_{32}\eta_2 + s_{33}\eta_3 \,. \end{aligned}$$

From this we see that $\mathrm{Var}(\varepsilon_1) = s_{11}^2$, $\mathrm{Var}(\varepsilon_2) = s_{21}^2 + s_{22}^2$, and $\mathrm{Var}(\varepsilon_3) = s_{31}^2 + s_{32}^2 + s_{33}^2$. Also, $\mathrm{Cov}(\varepsilon_1, \varepsilon_2) = s_{11}s_{21}$, and so on. The elements ε_1 and ε_2 are correlated because of the common influence of η_1 on both of them. They are not perfectly correlated because η_2 enters ε_2 without affecting ε_1. Similar analysis applies to ε_1 and ε_3, and ε_2 and ε_3. Essentially, the Choleski factor expresses K correlated terms as arising from K independent components, with each component "loading" differently onto each term. For any pattern of covariance, there is some set of loadings from independent components that reproduces that covariance.

9.2.6 Accept-reject for truncated multivariate densities

The procedure in section 9.2.4 for drawing from truncated densities applies only to univariate distributions. With multivariate densities, drawing from a truncated support is more difficult. We describe an accept-reject procedure that can always be applied. However, as we will see, there are disadvantages of the approach that might cause a researcher to choose another approach when possible.

Suppose we want to draw from multivariate density $g(\varepsilon)$ within the range $a \leq \varepsilon \leq b$ where a and b are vectors with the same length as ε. That is, we want to draw from $f(\varepsilon) = \frac{1}{k}g(\varepsilon)$ if $a \leq \varepsilon \leq b$, and $= 0$ otherwise , where k is the normalizing constant. We can obtain draws from f by simply drawing from g and retaining ("accepting") the draws that are within the relevant range and discarding ("rejecting") the draws that are outside the range. The advantage of this procedure is that it can be applied whenever it is possible to draw from the

untruncated density. Importantly, the normalizing constant, k, does not need to be known for the truncated density. This fact is useful since the normalizing constant is usually difficult to calculate.

The disadvantage of the procedure is that the number of draws that are accepted (that is, the number of draws from f that are obtained) is not fixed but rather is itself random. If R draws are taken from g, then the expected number of accepts is kR. This expected number is not known without knowing k, which, as stated, is usually difficult to calculate. It is therefore hard to determine an appropriate number of draws to take from g. More importantly, the actual number of accepted draws will generally differ from the expected number. In fact, there is a positive probability of obtaining no accepts from a fixed number of draws. When the truncation space is small (or, more precisely, when k is small), obtaining no accepts, and hence no draws from the truncated density, is a likely event.

This difficulty can be circumvented by drawing from g until a certain number of accepted draws is obtained. That is, instead of setting in advance the number of draws from g that will be taken, the researcher can set the number of draws from f that are obtained. Of course, the researcher will not know how long it will take to attain the set number.

In most situations, other procedures can be applied more easily to draw from a multivariate truncated density. Nevertheless, it is important to remember that, when nothing else seems possible with a truncated distribution, the accept-reject procedure can be applied.

9.2.7 Importance sampling

Suppose ε has a density $f(\varepsilon)$ that cannot be easily drawn from by the other procedures. Suppose further that there is another density, $g(\varepsilon)$, that can easily be drawn from. Draws from $f(\varepsilon)$ can be obtained as follows. Take a draw from $g(\varepsilon)$ and label it ε^1. Weight the draw by $f(\varepsilon^1)/g(\varepsilon^1)$. Repeat this process many times. The set of weighted draws is equivalent to a set of draws from f.

To verify this fact, we show that the cumulative distribution of the weighted draws from g is the same as the cumulative distribution of draws from f. Consider the share of draws from g that are below some

value m, with each draw weighted by f/g. This share is:

$$\begin{aligned}\int \left[\frac{f(\varepsilon)}{g(\varepsilon)}\right] I(\varepsilon < m) g(\varepsilon)\, d\varepsilon &= \int_{-\infty}^{m} \left[\frac{f(\varepsilon)}{g(\varepsilon)}\right] g(\varepsilon)\, d\varepsilon \\ &= \int_{-\infty}^{m} f(\varepsilon)\, d\varepsilon = F(m)\,.\end{aligned}$$

In simulation, draws from a density are used to calculate the average of a statistic over that density. Importance sampling can be seen as a change in the statistic and a corresponding change in the density that makes the density easy to draw from. Suppose we want to calculate $\int t(\varepsilon) f(\varepsilon)\, d\varepsilon$, but find it hard to draw from f. We can multiply the integrand by $g \div g$ without changing its value, such that the integral is $\int t(\varepsilon)\frac{f(\varepsilon)}{g(\varepsilon)} g(\varepsilon)\, d\varepsilon$. To simulate the integral, we take draws from g, calculate $t(\varepsilon)[f(\varepsilon)/g(\varepsilon)]$ for each draw, and average the results. We have simply transformed the integral so that it is easier to simulate.

The density f is called the target density, and g is called the proposal density. The requirements for importance sampling are that (1) the support of $g(\varepsilon)$ needs to cover the support of f, so that any ε that could arise with f can also arise with g, and (2) the ratio $f(\varepsilon)/g(\varepsilon)$ must be finite for all values of ε, so that this ratio can always be calculated.

A useful illustration of importance sampling arises with multivariate truncated normals. Suppose we want to draw from $N(0, \Omega)$ but with each element being positive (i.e., truncated below at zero.) The density is

$$f(\varepsilon) = \frac{1}{k(2\pi)^{\frac{1}{2}K}|\Omega|^{\frac{1}{2}}} e^{(-\frac{1}{2}\varepsilon'\Omega^{-1}\varepsilon)}$$

for $\varepsilon \geq 0$ and 0 otherwise, where K is the dimension of ε and k is the normalizing constant. (We assume for the purposes of this example that k is known. In reality, calculating k might take simulation in itself.) Drawing from this density is difficult because the elements of ε are correlated as well as truncated. However, we can use the procedure in section 9.2.4 to draw independent truncated normals and then apply importance sampling to create the correlation. Draw K univariate normals truncated below at zero, using the procedure in section 9.2.4. These draws collectively constitute a draw of a K-dimensional vector ε from the positive quadrant support with density

$$g(\varepsilon) = \frac{1}{m(2\pi)^{\frac{1}{2}K}} e^{(-\frac{1}{2}\varepsilon'\varepsilon)}$$