Random Sample Generation and Simulation of Probit Choice Probabilities

Based on sections 9.1-9.2 and 5.6 of Kenneth Train’s *Discrete Choice Methods with Simulation*

Presented by Jason Blevins
Applied Microeconometrics Reading Group
Duke University

21 June 2006

“Anyone attempting to generate random numbers by deterministic means is, of course, living in a state of sin.”
—John Von Neumann, 1951
Outline

- Density simulation and sampling
  - Univariate
  - Truncated univariate
  - Multivariate Normal
  - Accept-Reject Method for truncated densities
  - Importance sampling
  - Gibbs sampling
  - The Metropolis-Hastings Algorithm
- Simulation of Probit Choice Probabilities
  - Accept-Reject Simulator
  - Smoothed AR Simulators
  - GHK Simulator
Simulation in Econometrics

- Goal: approximate a conditional expectation which lacks a closed form.
- Statistic of interest: \( t(\epsilon) \), where \( \epsilon \sim F \).
- Want to approximate \( \mathbb{E}[t(\epsilon)] = \int t(\epsilon)f(\epsilon)d\epsilon \).
- Basic idea: calculate \( t(\epsilon) \) for \( R \) draws of \( \epsilon \) and take the average.
  - Unbiased: \( \mathbb{E}\left[ \frac{1}{R} \sum_{r=1}^{R} t(\epsilon^r) \right] = \mathbb{E}[t(\epsilon)] \)
  - Consistent: \( \frac{1}{R} \sum_{r=1}^{R} t(\epsilon^r) \overset{p}{\to} \mathbb{E}[t(\epsilon)] \)
- This is straightforward if we can generate draws from \( F \).
- In discrete choice models we want to simulate the probability that agent \( n \) chooses alternative \( i \).
  - Utility: \( U_{n,j} = V_{n,j} + \epsilon_{n,j} \) with \( \epsilon_n \sim F(\epsilon_n) \).
  - \( B_{n,i} = \{\epsilon_n \mid V_{n,i} + \epsilon_{n,i} > V_{n,j} + \epsilon_{n,j} \ \forall j \neq i\} \).
  - \( P_{n,i} = \int 1_{B_{n,i}}(\epsilon_n) f(\epsilon_n)d\epsilon_n \).
Random Number Generators

- True Random Number Generators:
  - Collect entropy from system (keyboard, mouse, hard disk, etc.)
  - Unix: /dev/random, /dev/urandom
- Pseudo-Random Number Generators:
  - Linear Congruential Generators \((x_{n+1} = ax_n + b \mod c)\): fast but predictable, good for Monte Carlo
  - Nonlinear: more difficult to determine parameters, used in cryptography
- Desirable properties for Monte Carlo work:
  - Portability
  - Long period
  - Computational simplicity
- DIEHARD Battery of Tests of Randomness, Marsaglia (1996)
Uniform and Standard Normal Generators

- Canned:
  - Matlab: `rand()`, `randn()`
  - Stata: `uniform()`, `invnormal(uniform())`

- Known algorithms:
  - Box-Muller algorithm
  - Marsaglia and Zaman (1994): `mzran`
  - Numerical Recipes, Press et al. (2002): `ran1`, `ran2`, `ran3`, `gasdev`
Simulating Univariate Distributions

- Direct vs. indirect methods.
- Transformation
  - Let $u \sim N(0, 1)$. Then $v = \mu + \sigma u \sim N(\mu, \sigma^2)$ and
  - $w = e^{\mu + \sigma u} \sim \text{Lognormal}(\mu, \sigma^2)$.
- Inverse CDF transformation:
  - Let $u \sim N(0, 1)$. If $F(\epsilon)$ is invertible, then $\epsilon = F^{-1}(u) \sim F(\epsilon)$.
  - Only works for univariate distributions
Figure 9.1. Draw of $\mu^1$ from uniform and create $\varepsilon^1 = F^{-1}(\mu)$. 
Truncated Univariate Distributions

- Want to draw from $g(\epsilon \mid a \leq \epsilon \leq b)$.
- Conditional density in terms of unconditional distribution $f(\epsilon)$:

$$g(\epsilon \mid a \leq \epsilon \leq b) = \begin{cases} \frac{f(\epsilon)}{F(b)-F(a)}, & \text{if } a \leq \epsilon \leq b \\ 0, & \text{otherwise} \end{cases}$$

- Drawing is analogous to using the inverse CDF transformation.
- Let $\mu \sim \mathcal{U}(0,1)$ and define $\bar{\mu} = (1-\mu)F(a) + \mu F(b)$. $\epsilon = F^{-1}(\bar{\mu})$ is necessarily between $a$ and $b$. 
Figure 9.2. Draw of $\bar{\mu}^1$ between $F(a)$ and $F(b)$ gives draw $\varepsilon^1$ from $f(\varepsilon)$ between $a$ and $b$. 
The Multivariate Normal Distribution

- Assuming we can draw from $N(0, 1)$, we can generate draws from any multivariate normal distribution $N(\mu, \Omega)$.
- Let $LL^T$ be the Cholesky decomposition of $\Omega$ and let $\eta \sim N(0, I)$.
- Then, since a linear transformation of a Normal r.v. is also Normal:

$$\epsilon = \mu + L\eta \sim N(\mu, \Omega)$$

$$\mathbb{E} [\epsilon] = \mu + L\mathbb{E} [\eta] = \mu$$

$$\text{Var}(\epsilon) = \mathbb{E} [(L\eta)(L\eta)^T]$$
$$= \mathbb{E} [L\eta\eta^TL^T]$$
$$= L\mathbb{E} [\eta\eta^T] L^T$$
$$= L\text{Var}(\eta) L^T = \Omega$$
The Accept-Reject Method for Truncated Densities

- Want to draw from a multivariate density $g(\epsilon)$, but truncated so that $a \leq \epsilon \leq b$ with $a, b, \epsilon \in \mathbb{R}^l$.
- The truncated density is $f(\epsilon) = \frac{1}{k}g(\epsilon)$ for some normalizing constant $k$.
- Accept-Reject method:
  - Draw $\epsilon'$ from $f(\epsilon)$.
  - Accept if $a \leq \epsilon' \leq b$, reject otherwise.
  - Repeat for $r = 1, \ldots, R$.
- Accept on average $kR$ draws.
- If we can draw from $f$, then we can draw from $g$ without knowing $k$.
- Disadvantages:
  - Size of resulting sample is random if $R$ is fixed.
  - Hard to determine required $R$.
  - Positive probability that no draws will be accepted.
- Alternatively, fix the number of draws to accept and repeat until satisfied.
Importance Sampling

- Want to draw from $f$ but drawing from $g$ is easier.
- Transform the target expectation into an integral over $g$:
  \[
  \int t(\epsilon)f(\epsilon)d\epsilon = \int t(\epsilon)\frac{f(\epsilon)}{g(\epsilon)}g(\epsilon)d\epsilon.
  \]

- Importance Sampling: Draw $\epsilon^r$ from $g$ and weight by $\frac{f(\epsilon^r)}{g(\epsilon^r)}$.
- The weighted draws constitute a sample from $f$.
- The support of $g$ must cover that of $f$ and $\sup \frac{f}{g}$ must be finite.
- To show equivalence, consider the CDF of the weighted draws:
  \[
  \int \frac{f(\epsilon)}{g(\epsilon)}1(\epsilon < m)g(\epsilon)d\epsilon = \int_{-\infty}^{m} \frac{f(\epsilon)}{g(\epsilon)}g(\epsilon)d\epsilon
  \]
  \[
  = \int_{-\infty}^{m} f(\epsilon)d\epsilon = F(m)
  \]
The Gibbs Sampler

- Used when it is difficult to draw from a joint distribution but easy to draw from the conditional distribution.

- Consider a bivariate case: \( f(\epsilon_1, \epsilon_2) \).

- Drawing iteratively from conditional densities converges to draws from the joint distribution.

- The Gibbs Sampler: Choose an initial value \( \epsilon_1^0 \).
  - Draw \( \epsilon_2^0 \sim f_2(\epsilon_2 | \epsilon_1^0) \), \( \epsilon_1^1 \sim f_1(\epsilon_1 | \epsilon_2^0) \), \ldots, \( \epsilon_1^t \sim f_1(\epsilon_1 | \epsilon_2^{t-1}) \), \( \epsilon_2^t \sim f_2(\epsilon_2 | \epsilon_1^t) \).
  - The sequence of draws \( \{(\epsilon_1^0, \epsilon_2^0), \ldots, (\epsilon_1^t, \epsilon_2^t)\} \) converges to draws from \( f(\epsilon_1, \epsilon_2) \).

The Gibbs Sampler: Example

- $\epsilon_1, \epsilon_2 \sim N(0, 1)$.
- Truncation: $\epsilon_1 + \epsilon_2 \leq m$.
- Ignoring truncation, $\epsilon_1 | \epsilon_2 \sim N(0, 1)$.
- Truncated univariate sampling:

$$\begin{align*}
\mu & \sim U(0, 1) \\
\tilde{\mu} & = (1 - \mu)\Phi(0) + \mu\Phi(m - \epsilon_2) \\
\epsilon_1 & = \Phi^{-1}(\mu\Phi(m - \epsilon_2))
\end{align*}$$

Figure 9.3. Truncated normal density.
The Metropolis-Hastings Algorithm

- Only requires being able to evaluate $f$ and draw from $g$.

- **Metropolis-Hastings Algorithm:**
  1. Let $\epsilon^0$ be some initial value.
  2. Choose a trial value $\tilde{\epsilon}^1 = \epsilon^0 + \eta$, $\eta \sim g(\eta)$, where $g$ has zero mean.
  3. If $f(\tilde{\epsilon}^1) > f(\epsilon^0)$, accept $\tilde{\epsilon}^1$.
  4. Otherwise, accept $\tilde{\epsilon}^1$ with probability $f(\tilde{\epsilon}^1)/f(\epsilon^0)$.
  5. Repeat for many iterations.

- The sequence $\{\epsilon^t\}$ converges to draws from $f$.

- Useful for sampling truncated densities when the normalizing factor is unknown.

- Description of algorithm: Chib and Greenberg (1995)
Calculating Probit Choice Probabilities

- **Probit Model:**
  - Utility: \( U_{n,j} = V_{n,j} + \epsilon_{n,j} \) with \( \epsilon_n \sim N(0, \Omega) \).
  - \( B_{n,i} = \{ \epsilon_n \mid V_{n,i} + \epsilon_{n,i} > V_{n,j} + \epsilon_{n,j}, \forall j \neq i \} \).
  - \( P_{n,i} = \int_{B_{n,i}} \phi(\epsilon_n) d\epsilon_n \).

- **Non-simulation methods:**
  - Quadrature: approximate the integral using a specifically chosen set of evaluation points and weights (Geweke, 1996, Judd, 1998).
  - Clark algorithm: maximum of several normal r.v. is itself approximately normal (Clark, 1961, Daganzo et al., 1977).

- **Simulation methods:**
  - Accept-reject method
  - Smoothed accept-reject
  - GHK (Geweke-Hajivassiliou-Keane)
The Accept-Reject Simulator

- Straightforward:
  1. Draw from distribution of unobservables.
  2. Determine the agent’s preferred alternative.
  3. Repeat $R$ times.
  4. The simulated choice probability for alternative $i$ is the proportion of times
     the agent chooses alternative $i$.

- General:
  - Applicable to any discrete choice model.
  - Works with any distribution that can be drawn from.
The Accept-Reject Simulator for Probit

- Let $B_{n,i} = \{\epsilon_n | V_{n,i} + \epsilon_{n,i} > V_{n,j} + \epsilon_{n,j}, \ \forall j \neq i\}$. The Probit choice probabilities are:
  $$P_{n,i} = \int \mathbb{1}_{B_{n,i}} (\epsilon_n) \phi(\epsilon_n) d\epsilon_n.$$ 

- Accept-Reject Method:
  1. Take $R$ draws $\{\epsilon^1_n, \ldots, \epsilon^R_n\}$ from $\mathcal{N}(0, \Omega)$ using the Cholesky decomposition $LL^\top = \Omega$ to transform iid draws from $\mathcal{N}(0, 1)$.
  2. Calculate the utility for each alternative: $U^r_{n,j} = V_{n,j} + \epsilon^r_{n,j}$.
  3. Let $d^r_{n,j} = 1$ if alternative $j$ is chosen and zero otherwise.
  4. The simulated choice probability for alternative $i$ is:
  $$\hat{P}_{n,i} = \frac{1}{R} \sum_{r=1}^R d^r_{n,i}$$
The Accept-Reject Simulator: Evaluation

- Main advantages: simplicity and generality.
- Can also be applied to the error differences in discrete choice models.
  - Slightly faster
  - Conceptually more difficult
- Disadvantages:
  - $\hat{P}_{n,i}$ will be zero with positive probability.
  - $\hat{P}_{n,i}$ is a step function and the simulated log-likelihood is not differentiable.
  - Gradient methods are likely to fail (gradient is either 0 or undefined).
Figure 5.1. The AR simulator is a step function in parameters.
The Smoothed Accept-Reject Simulator

- Replace the indicator function with a general function of $U_{n,j}$ for $j = 1, \ldots, J$ that is:
  - increasing in $U_{n,i}$ and decreasing in $U_{n,j}$ for $j \neq i$,
  - strictly positive, and
  - twice differentiable.
- McFadden (1989) suggested the Logit-smoothed AR simulator:
  1. Draw $\epsilon^r_n \sim \mathcal{N}(0, \Omega)$, for $r = 1, \ldots, R$.
  2. Calculate $U^r_{n,j} = V_{n,j} + \epsilon^r_{n,j} \quad \forall j, r$.
  3. Calculate the smoothed choice function for each simulation to find $\hat{P}_{n,i}$:

$$S^r_i = \frac{\exp(U^r_{n,i}/\lambda)}{\sum_{j=1}^J \exp(U^r_{n,j}/\lambda)},$$

$$\hat{P}_{n,i} = \frac{1}{R} \sum_{r=1}^R S^r_i$$
Figure 5.2. AR smoother.
The Smoothed Accept-Reject Simulator: Evaluation

- Simulated log-likelihood using smoothed choice probabilities is... smooth.
- Slightly more difficult to implement than AR simulator.
- Can provide a behavioral interpretation.
- Choice of smoothing parameter $\lambda$ is arbitrary.
- Objective function is modified.
- Use alternative optimization methods instead (simulated annealing)?
The GHK Simulator

- GHK: Geweke, Hajivassiliou, Keane.
- Simulates the Probit model in differenced form.
- For each $i$, simulation of $P_{n,i}$ uses utility differences relative to $U_{n,i}$.
- Basic idea: write the choice probability as a product of conditional probabilities.
- We are much better at simulating univariate integrals over $N(0, 1)$ than those over multivariate normal distributions.
GHK with Three Alternatives

- An example with three alternatives:

\[ U_{n,j} = V_{n,j} + \epsilon_{n,j}, \quad j = 1, 2, 3 \quad \text{with} \quad \epsilon_n \sim \mathcal{N}(0, \Omega) \]

- Assume \( \Omega \) has been normalized for identification.

- Consider \( P_{n,1} \). Difference with respect to \( U_{n,1} \):

\[ \tilde{U}_{n,j,1} = \tilde{V}_{n,j,1} + \tilde{\epsilon}_{n,j,1}, \quad j = 2, 3 \quad \text{with} \quad \tilde{\epsilon}_{n,1} \sim \mathcal{N}(0, \tilde{\Omega}_1) \]

\[ P_{n,1} = \mathbb{P}\left( \tilde{U}_{n,2,1} < 0, \tilde{U}_{n,3,1} < 0 \right) = \mathbb{P}\left( \tilde{V}_{n,2,1} + \tilde{\epsilon}_{n,2,1} < 0, \tilde{V}_{n,3,1} + \tilde{\epsilon}_{n,3,1} < 0 \right) \]

- \( P_{n,1} \) is still hard to evaluate because \( \tilde{\epsilon}_{n,j,1} \)'s are correlated.
GHK with Three Alternatives

• One more transformation. Let $L_1 L_1^\top$ be the Cholesky decomposition of $\tilde{\Omega}_1$:

$$L_1 = \begin{pmatrix} c_{aa} & 0 \\ c_{ab} & c_{bb} \end{pmatrix}$$

• Then we can express the errors as:

$$\tilde{\epsilon}_{n,2,1} = c_{aa} \eta_1$$

$$\tilde{\epsilon}_{n,3,1} = c_{ab} \eta_1 + c_{bb} \eta_2$$

where $\eta_1, \eta_2$ are iid $N(0, 1)$.

• The differenced utilities are then

$$\tilde{U}_{n,2,1} = \tilde{V}_{n,2,1} + c_{aa} \eta_1$$

$$\tilde{U}_{n,3,1} = \tilde{V}_{n,3,1} + c_{ab} \eta_1 + c_{bb} \eta_2$$
GHK with Three Alternatives

- $P_{n,1}$ is easier to simulate now:

$$P_{n,1} = \mathbb{P} \left( \tilde{V}_{n,2,1} + c_{aa}\eta_1 < 0, \tilde{V}_{n,3,1} + c_{ab}\eta_1 + c_{bb}\eta_2 < 0 \right)$$

$$= \mathbb{P} \left( \eta_1 < -\frac{\tilde{V}_{n,2,1}}{c_{aa}} \right) \mathbb{P} \left( \eta_2 < -\frac{\tilde{V}_{n,3,1} + c_{ab}\eta_1}{c_{bb}} \right) \left| \eta_1 < -\frac{\tilde{V}_{n,2,1}}{c_{aa}} \right)$$

$$= \Phi \left( -\frac{\tilde{V}_{n,2,1}}{c_{aa}} \right) \int_{-\infty}^{-\tilde{V}_{n,2,1}/c_{aa}} \Phi \left( -\frac{\tilde{V}_{n,3,1} + c_{ab}\eta_1}{c_{bb}} \right) \phi(\eta_1) d\eta_1$$

- First term only requires evaluating the standard Normal CDF.
- Integral is over a truncated univariate standard Normal distribution.
- The ‘statistic’ in this case is the standard Normal CDF.
Figure 5.3. Probability of alternative 1.
GHK with Three Alternatives: Simulation

\[
\Phi \left( -\frac{\tilde{V}_{n,1}^{2.1}}{c_{aa}} \right) \int_{-\infty}^{-\tilde{V}_{n,1}^{2.1}/c_{aa}} \Phi \left( -\frac{\tilde{V}_{n,1}^{3.1} + c_{ab}\eta_1}{c_{bb}} \right) \phi(\eta_1) d\eta_1 = k \int_{-\infty}^{\tilde{\eta}_1} t(\eta_1) \phi(\eta_1) d\eta_1
\]

1. Calculate \( k = \Phi \left( -\frac{\tilde{V}_{n,1}^{2.1}}{c_{aa}} \right) \).

2. Draw \( \eta_1^r \) from \( N(0, 1) \) truncated at \( -\tilde{V}_{n,1}^{2.1}/c_{aa} \) for \( r = 1, \ldots, R \): Draw \( \mu^r \sim U(0, 1) \) and calculate \( \eta_1^r = \Phi^{-1}\left( \mu^r \Phi \left( -\frac{\tilde{V}_{n,1}^{2.1}}{c_{aa}} \right) \right) \).

3. Calculate \( t^r = \Phi \left( -\frac{\tilde{V}_{n,1}^{3.1} + c_{ab}\eta_1^r}{c_{bb}} \right) \) for \( r = 1, \ldots, R \).

4. The simulated choice probability is \( \hat{P}_{n,1} = k \frac{1}{R} \sum_{r=1}^{R} t^r \)
Figure 5.4. Probability that $\eta_2$ is in the correct range, given $\eta_1^r$. 
GHK as Importance Sampling

\[ P_{n,1} = \int \mathbb{1}_B (\eta) \, g(\eta) \, d\eta \]

where \( B = \{ \eta \mid \tilde{U}_{n,j,i} < 0 \ \forall j \neq i \} \) and \( g(\eta) \) is the standard Normal PDF.

- Direct (AR) simulation involves drawing from \( g \) and calculating \( \mathbb{1}_B (\eta) \).
- GHK draws from a different density \( f(\eta) \) (the truncated normal):

\[
f(\eta) = \begin{cases} \frac{\phi(\eta_1)}{\Phi(-\tilde{V}_{n,1,i}/c_{11})} \frac{\phi(\eta_2)}{\Phi(-\tilde{V}_{n,2,i} + c_{21}\eta_1)/c_{22})} \cdots, & \text{if } \eta \in B \\ 0, & \text{otherwise} \end{cases}
\]

- Define \( \hat{P}_{i,n}(\eta) = \Phi(-\tilde{V}_{n,1,i}/c_{11})\Phi(-\tilde{V}_{n,2,i} + c_{21}\eta_1)/c_{22}) \cdots \).
- \( f(\eta) = g(\eta)/\hat{P}_{n,i}(\eta) \) on \( B \).
- \( P_{n,i} = \int \mathbb{1}_B (\eta) \, g(\eta) \, d\eta = \int \mathbb{1}_B (\eta) \, \frac{g(\eta)}{g(\eta)/\hat{P}_{i,n}(\eta)} \, f(\eta) \, d\eta = \int \hat{P}_{i,n}(\eta) \, f(\eta) \, d\eta \)
References


