Lecture Notes on Monte Carlo Methods
Fall Semester, 2005
Courant Institute of Mathematical Sciences, NYU
Jonathan Goodman, goodman@cims.nyu.edu

Chapter 2: Simple Sampling of Gaussians.
created August 26, 2005

Generating univariate or multivariate Gaussian random variables is simple and fast. There should be no reason ever to use approximate methods based, for example, on the Central limit theorem.

1 Box Muller

It would be nice to get a standard normal from a standard uniform by inverting the distribution function, but there is no closed form formula for this distribution function $N(x) = P(X < x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-x'^2/2} dx'$. The Box Muller method is a brilliant trick to overcome this by producing two independent standard normals from two independent uniforms. It is based on the familiar trick for calculating

$$I = \int_{-\infty}^{\infty} e^{-x^2/2} dx \; .$$

This cannot be calculated by "integration" – the indefinite integral does not have an algebraic expression in terms of elementary functions (exponentials, logs, trig functions). However,

$$I^2 = \int_{-\infty}^{\infty} e^{-x^2/2} dx \int_{-\infty}^{\infty} e^{-y^2/2} dy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2+y^2)/2} dx dy \; .$$

The last integral can be calculated using polar coordinates $x = r\cos(\theta)$, $y = r\sin(\theta)$ with area element $dxdy = rdrd\theta$, so that

$$I^2 = \int r = 0^{\infty} \int_{\theta=0}^{2\pi} e^{-r^2/2} r dr d\theta = 2\pi \int r = 0^{\infty} e^{-r^2/2} r dr \; .$$

Unlike the original x integral, this r integral is elementary. The substitution $s = r^2/2$ gives ds = rdr and

$$I^2 = 2\pi \int_{s=0}^{\infty} e^{-s} ds = 2\pi \; .$$

The Box Muller algorithm is a probabilistic interpretation of this trick. If (X, Y) is a pair of independent standard normals, then the probability density is a product:

$$f(x, y) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \cdot \frac{1}{\sqrt{2\pi}} e^{-y^2/2} = \frac{1}{2\pi} e^{-(x^2+y^2)/2} \; .$$

Since this density is radially symmetric, it is natural to consider the polar coordinate random variables (R, Θ) defined by $0 \leq \Theta < 2\pi$ and $X = R\cos(\Theta)$, and $Y = R\sin(\Theta)$. Clearly Θ is uniformly distributed in the interval $[0, 2\pi]$ and may be sampled using

$$\Theta = 2\pi U_1 \ .$$

Unlike the original distribution function $N(x)$, there is a simple expression for the R distribution function:

$$G(R) = P(R \leq r) = \int_{r'=0}^{r} \int_{\Theta=0}^{2\pi} \frac{1}{2\pi} e^{-r'^2/2} r dr d\theta = \int_{r'=0}^{r} e^{-r'^2/2} r dr \ .$$

The same change of variable $r'^2/2 = s$, $r'dr' = ds$ (so that $r' = r$ when $s = r^2/2$) allows us to calculate

$$G(r) = \int_{s=0}^{r^2/2} e^{-s} dx = 1 - e^{-r^2/2} \ .$$

Therefore, we may sample R by solving the distribution function equation[1]

$$G(R) = 1 - e^{-R^2/2} = 1 - U_2 \ ,$$

whose solution is $R = \sqrt{-2\ln(U_2)}$. Altogether, the Box Muller method takes independent standard uniform random variables U_1 and U_2 and produces independent standard normals X and Y using the formulas

$$\Theta = 2\pi U_1 \ , \quad R = \sqrt{-2\ln(U_2)} \ , \quad X = R\cos(\Theta) \ , \quad Y = R\sin(\Theta) \ . \tag{1}$$

It may seem odd that X and Y in (13) are independent given that they use the same R and Θ. Not only does our algebra shows that this is true, but we can test the independence computationally, and it will be confirmed.

Part of this method was generating a point "at random" on the unit circle. We suggested doing this by choosing Θ uniformly in the interval $[0, 2\pi]$ then taking the point on the circle to be $(\cos(\Theta), \sin(\Theta))$. This has the possible drawback that the computer must evaluate the sine and cosine functions. Another way to do this[2] is to choose a point uniformly in the 2×2 square $-1 \leq x \leq 1$, $1 \leq y \leq 1$ then rejecting it if it falls outside the unit circle. The first accepted point will be uniformly distributed in the unit disk $x^2 + y^2 \leq 1$, so its angle will be random and uniformly distributed. The final step is to get a point on the unit circle $x^2 + y^2 = 1$ by dividing by the length.

The methods have equal accuracy (both are exact in exact arithmetic). What distinguishes them is computer performance (a topic discussed more in a later lecture, hopefully). The rejection method, with an acceptance probability $\frac{\pi}{4} \approx 78\%$, seems efficient, but rejection can break the instruction pipeline and slow a computation by a factor of ten. Also, the square root needed to compute

[1] Recall that $1 - U_2$ is a standard uniform if U_2 is.

[2] Suggested, for example, in the dubious book *Numerical Recipies*.

the length may not be faster to evaluate than sine and cosine. Moreover, the rejection method uses two uniforms while the Θ method uses just one.

The method can be reversed to solve another sampling problem, generating a random point on the "unit spnere" in R^n. If we generate n independent standard normals, then the vector $X = (X_1, \ldots, X_n)$ has all angles equally likely (because the probability density is $f(x) = \left(\frac{1}{\sqrt{2\pi}}\right)^n \exp(-(x_1^2 + \cdots + x_n^2)/2)$, which is radially symmetric. Therefore $X/\left\|X\right\|$ is uniformly distributed on the unit sphere, as desired.

1.1 Other methods for univariate normals

The Box Muller method is elegant and reasonably fast and is fine for casual computations, but it may not be the best method for hard core users. Many software packages have native standard normal random number generators, which (if they are any good) use expertly optimized methods. There is very fast and accurate software on the web for directly inverting the normal distribution function $N(x)$. This is particularly important for *quasi Monte Carlo*, which substitutes equidistributed sequences for random sequences (see a later lecture).

2 Multivariate normals

An n component multivariate normal, X, is characterized by its mean $\mu = E[X]$ and its covariance matrix $C = E[(X - \mu)(X - \mu)^t]$. We discuss the problem of generating such an X with mean zero, since we achieve mean μ by adding μ to a mean zero multivariate normal. The key to generating such an X is the fact that if Y is an m component mean zero multivariate normal with covariance D and $X = AY$, then X is a mean zero multivariate normal with covariance

$$C = E\left[XX^t\right] = E\left[AY\,(AY)^t\right] = AE\left[YY^t\right]A^t = ADA^t\ .$$

We know how to sample the n component multivariate normal with $D = I$, just take the components of Y to be independent univariate standard normals. The formula $X = AY$ will produce the desired covariance matrix if we find A with $AA^t = C$.

A simple way to do this in practice is to use the *Choleski decomposition* from numerical linear algebra. This is a simple algorithm that produces a lower triangular matrix, L, so that $LL^t = C$. It works for any positive definite C.

In physical applications it is common that one has not C but its inverse, H. This would happen, for example, if X had the Gibbs-Boltzmann distribution with $kT = 1$ (it's easy to change this) and energy $\frac{1}{2}X^tHX$, and probability density $\frac{1}{Z}\exp(-\frac{1}{2}X^tHX)$. In large scale physical problems it may be impractical to calculate and store the covariance matrix $C = H^{-1}$ though the Choleski factorization $H = LL^t$ is available. Note that[3] $H^{-1} = L^{-t}L^{-1}$, so the choice

[3]It is traditional to write L^{-t} for the transpose of L^{-1}, which also is the inverse of L^t.

$A = L^{-t}$ works. Computing $X = L^{-t}Y$ is the same as solving for X in the equation $Y = L^t X$, which is the process of *back substitution* in numerical linear algebra.

In some applications one knows the eigenvectors of C (which also are the eigenvectors of H), and the corresponding eigenvalues. These (either the eigenvectors or the eigenvectors and eigenvalues) sometimes are called *principal components*. Let q_j be the eigenvectors, normalized to be orthonormal, and σ_j^2 the corresponding eigenvalues of C, so that

$$Cq_j = \sigma_j^2 q_j \ , \quad q_j^t q_k = \delta_{jk} \ .$$

Denote the q_j component of X by $Z_j = q_j^t X$. This is a linear function of X and therefore Gaussian with mean zero. It's variance (note: $Z_j = Z_j^t = X^t q_j$) is

$$E[Z_j^2] = E[Z_j \cdot Z_j^t] = q_j^t E[XX^t] q_j = q_j^t C q_j = \sigma_j^2 \ .$$

A similar calculation shows that Z_j and Z_k are uncorrelated and hence (as components of a multivariate normal) independent. Therefore, we can generate Y_j as independent standard normals and sample the Z_j using

$$Z_j = \sigma_j Y_j \ . \tag{2}$$

After that, we can get an X using

$$X = \sum_{j=1}^{n} Z_j q_j \ . \tag{3}$$

We restate this in matrix terms. Let Q be the orthogonal matrix whose columns are the orthonormal eigenvectors of C, and let Σ^2 be the diagonal matrix with σ_j^2 in the (j,j) diagonal position. The eigenvalue/eigenvector relations are

$$CQ = Q\Sigma^2 \ , \ \ Q^t Q = I = QQ^t \ . \tag{4}$$

The multivariate normal vector $Z = Q^t X$ then has covariance matrix $E[ZZ^t] = E[Q^t X X^t Q] = Q^t C Q = \Sigma^2$. This says that the Z_j, the components of Z, are independent univariate normals with variances σ_j^2. Therefore, we may sample Z by choosing its components by (14) and then reconstruct X by $X = QZ$, which is the same as (15). Alternatively, we can calculate, using (17) that

$$C = Q\Sigma^2 Q^t = Q\Sigma\Sigma Q^t = (Q\Sigma)\,(Q\Sigma)^t \ .$$

Therefore $A = Q\Sigma$ satisfies $AA^t = C$ and $X = AY = Q\Sigma Y = QZ$ has covariance C if the components of Y are independent standard univariate normals or the components of Z are independent univariate normals with variance σ_j^2.

3 Brownian motion examples

We illustrate these ideas for various kids of Brownian motion. Let $X(t)$ be a Brownian motion path. Choose a *final time* t and a *time step* $\Delta t = T/n$. The

observation times will be $t_j = j\Delta t$ and the *observations* (or *observation values*) will be $X_j = X(t_j)$. These observations may be assembled into a vector $X = (X_1, \ldots, X_n)^t$. We seek to generate sample observation vectors (or observation *paths*). How we do this depends on the *boundary conditions*.

The simplest case is *standard* Brownian motion. Specifying $X(0) = 0$ is a *Dirichlet* boundary condition at $t = 0$. Saying nothing about $X(T)$ is a *free* (or *Neumann*) condition at $t = T$. The joint probability density for the observation vector, $f(x) = f(x_1, \ldots, x_n)$, is found by multiplying the conditional densities. Given $X_k = X(t_k)$, the next observation $X_{k+1} = X(t_k + \Delta t)$ is Gaussian with mean X_k and variance Δt, so its conditional density is

$$\frac{1}{\sqrt{2\pi\Delta t}} e^{-(x_{k+1}-X_k)^2/2\Delta t} .$$

Multiply these together and use $X_0 = 0$ and you find (with the convention $x_0 = 0$)

$$f(x_1, \ldots, x_n) = \left(\frac{1}{2\pi\Delta t}\right)^{n/2} \exp\left(\frac{-1}{2 - Deltat} \sum_{k=0}^{n-1} (x_{k+1} - x_k)^2\right) . \qquad (5)$$

3.1 The random walk method

The simplest and possibly best way to generate a sample observation path, X, comes from the derivation of (1). First generate $X_1 = X(\Delta t)$ as a mean zero univariate normal with mean zero and variance Δt, i.e. $X_1 = \sqrt{\Delta t} Y_1$. Given X_1, X_2 is a univariate normal with mean X_1 and variance Δt, so we may take $X_2 = X_1 + \sqrt{\Delta t} Y_2$, and so on. This is the *random walk* method. If you just want to make standard Brownian motion paths, stop here. We push on for pedigogical purposes and to develop strategies that apply to other types of Brownian motion.

We describe the random walk method in terms of the matrices above, starting by identifying the matrices C and H. Examining (1) leads to

$$H = \frac{1}{\Delta t} \begin{pmatrix} 2 & -1 & 0 & & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & \vdots \\ 0 & -1 & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & 2 & -1 & 0 \\ & & \ddots & -1 & 2 & -1 \\ 0 & \cdots & & 0 & -1 & 1 \end{pmatrix}$$

This is a *tridiagonal* matrix with pattern $-1, 2, -1$ except at the bottom right corner. One can calculate the covariances C_{jk} from the random walk representation

$$X_k = \sqrt{\Delta t} \, (Y_1 + \cdots + Y_k) \ .$$

Since the Y_j are independent, we have

$$C_{kk} = \mathrm{var}(X_k) = \Delta t \cdot k \cdot \mathrm{var}(Y_j) = t_k \ ,$$

and, supposing $j < k$,

$$\begin{aligned} C_{jk} &= E[X_j X_k] \\ &= \Delta t E\left[((Y_1 + \cdots + Y_j) + (Y_{j+1} + \cdots + Y_k)) \cdot (Y_1 + \cdots + Y_j)\right] \\ &= \Delta t E\left[(Y_1 + \cdots + Y_j)^2\right] \\ &= t_j \ . \end{aligned}$$

These combine into the familiar formula

$$C_{jk} = \mathrm{cov}(X(t_j), X(t_k)) = \min(t_j, t_k) \ .$$

This is the same as saying that the matrix C is

$$C = \Delta t \begin{pmatrix} 1 & 1 & \cdots & & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ \vdots & 2 & 3 & \cdots & 3 \\ & \vdots & \vdots & & \vdots \\ 1 & 2 & 3 & \cdots & \ddots \end{pmatrix} \tag{6}$$

The random walk method for generating X may be expresses as

$$\begin{pmatrix} X_1 \\ \\ \vdots \\ \\ X_n \end{pmatrix} = \sqrt{\Delta t} \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & 0 & \vdots \\ \vdots & \vdots & & \ddots & \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} Y_1 \\ \\ \vdots \\ \\ Y_n \end{pmatrix} \ .$$

Thus, $X = AY$ with

$$A = \sqrt{\Delta t} \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & 0 & \vdots \\ \vdots & \vdots & & \ddots & \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix} \ . \tag{7}$$

The reader should do the matrix multiplication to check that indeed $C = AA^t$ for (6) and (7). Notice that H is a *sparse* matrix indicating *short range interactions* while C is *full* indicating *long range correlations*. This is true of in great number of physical applications, though it is rare to have an explicit formula for C.

We also can calculate the Choleski factorization of H. The reader can convince herself or himself that the Choleski factor, L, is *bidiagonal*, with nonzeros only on or immediately below the diagonal.

However, the formulas are simpler if we reverse the order of the coordinates. Therefore we define the coordinate reversed observation vector

$$\widetilde{X} = (X_n, x_{n-1}, \ldots, X_n)^t$$

and whose covariance matrix is

$$\widetilde{C} = \begin{pmatrix} t_n & t_{n-1} & \cdots & t_1 \\ t_{n-1} & t_{n-1} & & t_1 \\ \vdots & & \ddots & \\ t_1 & \cdots & & t_1 \end{pmatrix} ,$$

and energy matrix

$$\widetilde{H} = \frac{1}{\Delta t} \begin{pmatrix} 1 & -1 & 0 & & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & \vdots \\ 0 & -1 & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & 2 & -1 & 0 \\ & & \ddots & -1 & 2 & -1 \\ 0 & \cdots & & 0 & -1 & 2 \end{pmatrix} .$$

We seek the Choleski factorization $\widetilde{H} = \widetilde{L}\widetilde{L}^t$ with bidiagonal

$$\widetilde{L} = \frac{1}{\sqrt{\Delta t}} \begin{pmatrix} l_1 & 0 & \cdots \\ m_2 & l_2 & 0 \\ 0 & m_3 & \ddots \\ \vdots & \ddots & \ddots \end{pmatrix} .$$

Multiplying out $\widetilde{H} = \widetilde{L}\widetilde{L}^t$ leads to equations that successively determine the l_k and m_k:

$$\begin{aligned} l_1^2 &= 1 \implies l_1 = 1 , \\ l_1 m_2 &= -1 \implies m_2 = -1 , \\ l_1^2 + l_2^2 &= 2 \implies l_2 = 1 , \\ l_2 m_3 &= 1 \implies m_3 = -1 , \text{ etc.,} \end{aligned}$$

The result is $\widetilde{H} = \widetilde{L}\widetilde{L}^t$ with $\widetilde{L}$ simply

$$\widetilde{L} = \frac{1}{\sqrt{\Delta t}} \begin{pmatrix} 1 & 0 & \cdots \\ -1 & 1 & 0 \\ 0 & -1 & \ddots \\ \vdots & \ddots & \ddots \end{pmatrix} .$$

The sampling algorithm using this information is to find $\widetilde{X}$ from $\widetilde{Y}$ by solving $\widetilde{Y} = \widetilde{L}^t \widetilde{X}$:

$$\begin{pmatrix} Y_n \\ Y_{n-1} \\ \\ \vdots \\ \\ Y_1 \end{pmatrix} = \frac{1}{\sqrt{\Delta t}} \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & & \vdots \\ & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & -1 \\ 0 & \cdots & & 0 & 1 \end{pmatrix} \begin{pmatrix} X_n \\ X_{n-1} \\ \\ \vdots \\ \\ X_1 \end{pmatrix}$$

Solving from the bottom up (back substitution), we have

$$\begin{aligned} Y_1 &= \frac{1}{\sqrt{\Delta t}} X_1 \implies X_1 = \sqrt{\Delta t} Y_1 \, , \\ Y_2 &= \frac{1}{\sqrt{\Delta t}} (X_2 - X_1) \implies X_2 = X_1 + \sqrt{\Delta t} Y_2 \, , \text{ etc.} \end{aligned}$$

This whole process turns out to give the same random walk sampling method.

Had we not gone to the time reversed ($\widetilde{X}$, etc.) variables, we could have calculated the bidiagonal Choleski factor L numerically. This works for any problem with a tridiagonal energy matrix H and has a name in the control theory/estimation literature that escapes me. In particular, it will allow to find sample Brownian motion paths with other boundary conditions.

3.2 The Brownian bridge construction

The Brownian bridge construction is useful in the mathematical theory of Brownian motion. It also is the basis for the success of quasi Monte Carlo methods in finance. Suppose n is a power of 2: $n = 2^L$. We will construct the observation path X through a sequence of L refinements. First, notice that X_n is a univariate normal with mean zero and variance T, so we may take (with $Y_{k,l}$ being independent standard normals)

$$X_n = \sqrt{T} Y_{1,1} \, .$$

Given the value of X_n, the midoint observation, $X_{n/2}$, is a univariate normal[4] with mean $\frac{1}{2} X_n$ and variance $T/4$, so we may take

$$X_{\frac{n}{2}} = \frac{1}{2} X_n + \frac{\sqrt{T}}{2} Y_{2,1} \, .$$

At the first level, we chose the endpoint value for X. We could draw a first level path by connenting X_n to zero with a straight line. At the second level, or first refinement, we created a midpoint value. The second level path could be piecewise linear, connecting 0 to $X_{\frac{n}{2}}$ to X_n.

[4] We assign this and related claims below as exercises for the student.

The second refinement level creates values for the "quarter points". Given $X_{\frac{n}{2}}$, $X_{\frac{n}{4}}$ is a normal with mean $\frac{1}{2}X_{\frac{n}{2}}$ and variance $\frac{1}{4}\frac{T}{2}$. Similarly, $X_{\frac{3n}{4}}$ is a normal with mean $\frac{1}{2}(X_{\frac{n}{2}} + X_n)$ and variance $\frac{1}{4}\frac{T}{2}$. Therefore, we may take

$$X_{\frac{n}{4}} = \frac{1}{2}X_{\frac{n}{2}} + \frac{1}{2}\sqrt{\frac{T}{2}}Y_{3,1}$$

and

$$X_{\frac{3n}{4}} = \frac{1}{2}(X_{\frac{n}{2}} + X_n) + \frac{1}{2}\sqrt{\frac{T}{2}}Y_{3,2} \ .$$

The level three path would be piecewise linear with breakpoints at $\frac{1}{4}$, $\frac{1}{2}$, and $\frac{3}{4}$. Note that in each case we add a mean zero normal of the appropriate variance to the linear interpolation value.

In the general step, we go from the level $k-1$ path to the level k paths by creating values for the midpoints of the level $k-1$ intervals. The level k observations are $X_{\frac{j}{2^{k-1}}}$. The values with even j are known from the previous level, so we need values for odd j. That is, we want to interpolate between the $j = 2m$ value and the $j = 2m+2$ value and add a mean zero normal of the appropriate variance:

$$X_{\frac{(2m+1)n}{2^{k-1}}} = \frac{1}{2}\left(X_{\frac{2mn}{2^{k-1}}} + X_{\frac{(2m+2)n}{2^{k-1}}}\right) + \frac{1}{2^{(k-2)/2}}\sqrt{\frac{T}{2}}Y_{m,k} \ .$$

The reader should check that the vector of standard normals $Y = (Y_{1,1}, Y_{2,1}, Y_{3,1}, Y_{3,2}, \ldots)^t$ indeed has $n = 2^L$ components. The value of this method for quasi Monte Carlo comes from the fact that the most important values that determine the large scale structure of X are the first components of Y. As we will see, the components of the Y vectors of quasi Monte Carlo have uneven quality, with the first components being the best.

3.3 Principle components

The principle component eigenvalues and eigenvectors for many types of Brownian motion are known in closed form. In many of these cases, the *Fast Fourier Transform* (FFT) algorithm leads to a reasonably fast sampling method. These FFT based methods are slower than random walk or Brownian bridge sampling for standard random walk, but they sometimes are the most efficient for *fractional* Brownian motion. They may be better than Brownian bridge sampling with quasi Monte Carlo (I'm not sure about this).

The eigenvectors of H are known[5] to have components ($q_{j,k}$ is the k^{th} component of eigenvector q_j.)

$$q_{j,k} = const \cdot \sin(\omega_j t_k) \ . \tag{8}$$

[5]See e.g. *Numerical Analysis* by Eugene Isaacson and Herbert Keller.

The n eigenvectors and eigenvalues then are determined by the allowed values of ω_j, which, in turn, are determined throught the boundary conditions. We can find σ_j^2 in terms of ω_j using the eigenvalue equation $Hq_j = \sigma_j^2 q_j$ evaluated at any of the *interior* components $1 < k < n$:

$$\frac{1}{\Delta t}\left[-\sin(\omega_j(t_k - \Delta t)) + 2\sin(\omega_j t_k) - \sin(\omega_j(t_k + \Delta t))\right] = \sigma_j^2 \sin(\omega_j t_k) \ .$$

Doing the math shown that the eigenvalue equation is satisfied and that

$$\sigma_j^2 = 2\frac{1 - \cos(\omega_j \Delta t)}{\Delta t} \ . \tag{9}$$

The eigenvalue equation also is satisfied at $k = 1$ because the form (8) automatically satisfies the boundary condition $q_{j,0} = 0$. This is why we used the sine and not the cosine.

Only special values ω_j give $q_{j,k}$ that satisfy the eigenvalue equation at the right boundary point $k = n$.