Monte Carlo Lecture Notes II,

Jonathan Goodman *
Courant Institute of Mathematical Sciences, NYU

January 29, 1997

1 Introduction to Direct Sampling

We think of the computer "random number generator" as an oracle, producing independent random variables, ξ_1, ξ_2, $\ldots$, each uniformly distributed in the interval $[0,1]$. Direct sampling methods are methods that use these ξ_k to produce independent random variables with other probability distributions. The notation $X \sim \rho(x)$ will mean that the random variable X has a probability density function $\rho(x)$. This means that

$$\text{Prob}\left[X \in A\right] = \int_A \rho(x)dx$$

for any (measurable) set, A. With this notation, X can be a one dimensional or a multidimensional random variable. If X is a random variable, we say that X_k "is a sample of X" if X_k has the same density X has. Direct sampling means producing independent samples of a given random variable.

In many cases, the most efficient sampling methods are not direct, but "dynamic", or iterative. Dynamic methods, such as the Metropolis algorithm, produce a sequence of samples that are not independent of each other and do not exactly have the ρ density. However, as $k \to \infty$, the density for X_k

*goodman@cims.nyu.edu, or http://www.math.nyu.edu/faculty/goodman,

converges to ρ. Moreover, as the offset $l \to \infty$, the samples X_k and X_{k+l} approach independence. This is similar to the situation in classical numerical analysis. Direct methods for solving equations, such as Gauss elimination, produce (in "exact arithmetic") the exact answer in a specific amount of computer time. Nevertheless, for very large systems of equations that arise from discretizing partial differential equations, iterative methods, such as Gauss Seidel or multigrid, are preferred.

Direct sampling methods that are entirely deterministic (given the assumed uniform random variables) are called "mapping methods". One of the most famous mapping methods is the Box Muller method for generating standard normals. Rejection methods are the other main family of direct sampling methods. They are interesting and useful, and they are precursors to one of the main dynamic sampling methods, the Metropolis method.

2 Direct Sampling by Mapping

Mapping methods are methods that make a random variable by applying deterministic operations (i.e. mappings) to other random variables. They are the closest Monte Carlo relatives to direct methods in numerical analysis; they produce an exact answer (exactly independent random variables exactly distributed by ρ) in a deterministic amount of time.

The term "mapping method" is often used to refer to the special case of a one dimensional random variable, X, given as a function of a single uniform random variable, ξ. Suppose that the function $x = \phi(t)$ is defined and monotone for t in the range $0 < t < 1$ and that $X = \phi(\xi)$. We need to determine the probability density function, $\rho(x)$, for X. This can be done using the applied mathematicians' definition of ρ (indeed, the definition used by the entire world outside of pure mathematics):

$$\rho(x)dx = \text{Prob}\left[X \in (x, x + dx)\right] \quad . \tag{1}$$

To use this, we suppose that a number, x is in the range of ϕ (otherwise, $\rho(x) = 0$). Since ϕ is monotone[1], there is a unique t with $x = \phi(t)$. Moreover, ϕ maps an interval of length dt around t to an interval of length dx around x, where $dx = \phi'(t)dt$. The probability that X lands in the interval of length

[1] This is the only place where the monotonicity assumption is used

dx around x is the same as the probability that ξ is in the interval of size dt around t. This leads to the formula

$$\rho(x) = 1/\phi'(t) \ , \quad \text{where} \quad x = \phi(t) \ . \tag{2}$$

This one dimensional mapping method can be explained in terms of the distribution function

$$F(x) = \text{Prob}\,[X < x] = \int_{-\infty}^{x} \rho(x')dx' \ .$$

This function is monotone increasing (or, at any rate, nondecreasing) and maps the real line to the interval $[0, 1]$. If $x = \phi(t)$ is the inverse of this mapping, then $X = \phi(\xi)$ gives $X \sim \rho$. Indeed, if $F(x) = t$, which is the same as $\phi(t) = x$, then $\text{Prob}[X < x] = \text{Prob}[\xi < t] = t = F(x)$, as claimed. Thus, if we can compute the indefinite integral $F(x)$ and solve the equation $F(x) = t$ to find x for given $t \in [0, 1]$, then we can sample from ρ. This is the case for the exponential random variable (see below).

If we don't have closed form expressions for F or ϕ, we can tabulate them. The work that it takes to make the table by numerical integration will be dwarfed by the time taken to generate thousands or millions of samples. Tabulation may not be practical when ρ depends on several parameters whose values are not known in advance or change from sample to sample.

2.1 The Exponential Random Variable

An exponential random variable with mean μ is a random variable with density

$$\rho(x) = \frac{1}{\mu}e^{-x/\mu} \ , \quad \text{if } x > 0, \text{ and } \rho(x) = 0 \text{ otherwise.} \tag{3}$$

If X is an exponential with mean 1 then $Y = \mu X$ is exponential with mean μ (check this), so we need only generate an exponential with mean 1. Exponential random variables are important partly because they are a good way to simulate a continuous time Markov process with discrete state space.

It is natural to think of an exponential random variable as the random amount of time one waits for something to happen. The defining property of the *exponential* waiting time is that it has no knowledge of the past (This is the Markov property.). If you have waited a time, t and the event has not happened ($T > t$), then it is as though you had not waited at all. If a light bulb failure has an exponential distribution, than any bulb that has not failed is good as new. Such models are used commercially, for example to predict the MTBF (mean time before failure) of hard disks. The manufacturer runs a number of drives and records the failures in a few months. These observations are used to fit an exponential probability density and determine an empirical parameter, μ, which then appears in the advertisements. If a company actually tested its drives until half of them failed (on the order of 4 years) before advertising them, it would miss the market completely.

The formula (2) is a consequence of the Markov property. Suppose $\rho(t)$ is the density function for a random variable, T, with the Markov property. The probability of breaking immediately within time dt is $\rho(0)dt$. The Markov property states that this is also the probability of breaking in time interval $(t, t+dt)$, given that it has lasted until time t. From the formula for conditional probability, this leads to

$$\begin{aligned} \rho(0)dt &= \text{Prob}\,[T \in (t, t+dt) \mid T > t] \\ &= \frac{\text{Prob}\,[T \in (t, t+dt) \;\text{ and }\; T > t]}{\text{Prob}\,[T > t]} \\ &= \frac{\rho(t)dt}{1 - F(t)} \ , \end{aligned}$$

where $F(t) = \int_{-\infty}^{t} \rho(t')dt'$ is the distribution function for ρ. But, $\rho(t) = -F'(t)$ and $F(0) = 0$, so we have the differential equation and boundary condition for F

$$\rho(0)\,(1 - F(t)) = -F'(t) \ , \quad F(0) = 0 \ ,$$

which leads to $F(t) = 1 - e^{-t/\mu}$ and then to (2).

The direct sampling method for the exponential random variable is easier to explain than the exponential random variable itself. If $X = -\log(\xi)$ and ξ is uniform, then (1) shows that X is exponential. Direct application of the probability distribution formalism discussed above leads to the formula $X = -\log(1-\xi)$, which also works, since $1-\xi$ is also a uniformly distributed random variable.

2.2 The Box Muller Algorithm for Normals

A Gaussian random variable (also called "normal") with mean μ and variance σ^2 is a random variable with density function

$$\frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/2\sigma^2} \quad .$$

The "standard normal" random variable is a Gaussian with mean 0 and variance 1.

The Box Muller method, which is a clever mapping method, makes two independent standard normal random variables from two independent uniforms. The trick is that X and Y are independent standard normals if and only if their joint density function is

$$\rho(x,y) = \frac{1}{2\pi}e^{-(x^2+y^2)/2} \quad . \tag{4}$$

If we write (X,Y) in polar coordinates, $X = R\cos(\Theta)$, $Y = R\sin(\Theta)$ then (X,Y) will have the density (3) if R and Θ have the joint density

$$\tilde{\rho}(r,\theta) = \frac{1}{2\pi}re^{-r^2/2} \quad ,$$

where, of course, θ is restricted to a range such as $0 < \theta < 2\pi$. This can be sampled by taking Θ to be uniformly distributed in the interval $[0,2\pi]$ (i.e. $\Theta = 2\pi\xi_1$), and R by the mapping formula $R = \sqrt{-2\log(\xi_2)}$.

The Box Muller algorithm makes independent standard normals in pairs. The first two uniforms, ξ_1, and ξ_2, make the first two standard normals, X_1, and X_2. Then ξ_3 and ξ_4 make X_3 and X_4, and so on. Many large Monte Carlo codes avoid repeated function calls to random number generators by generating large lists of random numbers (say, $10,000$) each call. The Box Muller can easily be used to fill list of standard normal random variables.

2.3 Green's Function for the Klein Gordon Operator

The Klein Gordon operator is

$$-\triangle + m^2 \quad .$$

Klein and Gordon used it (actually, it's square root) in an attempt to make a relativistic version of Schrödinger's wave equation in quantum mechanics. It has many applications, including Monte Carlo. The Green's function for the Klein Gordon operator, G(x), is defined to be the solution of

$$-\triangle G + m^2 G = \delta(x) \ . \tag{5}$$

The solution in 3 dimensions is $G(x) = \frac{1}{|x|}e^{-m|x|}$. The solution in 2 dimensions is a "modified Bessel function": $G(x) = K_0(m\,|x|)$. By integrating both sides of (4), and assuming that G decays rapidly for large $|x|$, we find that

$$\int G(x)dx = \frac{1}{m^2} \ .$$

The probability density function we want to sample is $\rho(x) = \frac{1}{m^2}G(x)$. We will shortly see that $G(x) > 0$ for all X. We may at first be discouraged since we don't even have a formula for ρ.

The trick for sampling ρ, and for many related density functions, is to find a suitable integral representation. Integral representations for special functions in mathematical physics can usually be found from physical properties of the definition of the function. In this case, consider a generalization of (4) to more general right hand sides and operators:

$$Au = f \ . \tag{6}$$

Here A represents the Klein Gordon operator, u the Green's function, and f the delta function. If A is positive definite (as it is in the Klein Gordon case), we solve (5) using the dynamical system

$$\dot{v} = -Av \ , \quad v(0) = f \ . \tag{7}$$

If $v(t) \to 0$ quickly enough as $t \to \infty$, then

$$u = \int_{t=0}^{\infty} v(t)dt \tag{8}$$

satisfies (5). In the case of the Klein Gordon operator, (6) becomes

$$\partial_t v = \triangle v - m^2 v \ , \quad v(x,0) = \delta(x) \ . \tag{9}$$

If the m^2 term were not present, the solution would be given by the fundamental solution of the heat equation in d dimensions, namely

$$\frac{1}{(4\pi t)^{d/2}} e^{-|x|^2/4t} \quad .$$

The "decay term", $-m^2 v$, is handled by including an additional exponential decay factor. Therefore, the solution to (8) is

$$v(x,t) = \frac{1}{(4\pi t)^{d/2}} e^{-|x|^2/4t} e^{-m^2 t} \quad .$$

Finally, we can take the integral (7) to get the desired integral representation for G, and therefore (after adding a factor $1/m^2$) ρ:

$$\rho(x) = \int_{t=0}^{\infty} \frac{dt}{m^2} e^{-m^2 t} \cdot \frac{1}{(4\pi t)^{d/2}} e^{-|x|^2/4t} \quad . \tag{10}$$

The integral representation (9) suggests a strategy for sampling ρ. Notice that the first factor on the right is an exponential density with mean $1/m^2$ while the second is a gaussian density in d dimensions with each component having variance $2t$. The direct sampling algorithm is now: first, pick a random time, T, from the exponential density function with mean $1/m^2$ using the log mapping above; second, pick $X = (X_1, \ldots, X_d)$ by taking Y_k to be independent standard normal random variables made using the Box Muller algorithm, and $X_k = \sqrt{2T} Y_k$.

In this sampling algorithm, the random variable, T is not reported to the user, but it makes the method work. We have turned a seemingly hard d dimensional sampling problem into a much easier $d+1$ dimensional sampling problem. Some of the most effective innovative Monte Carlo methods developed in recent years, among them umbrella sampling and cluster algorithms, are based on clever enlargements of the sampling space.

3 Sampling by Rejection

Rejection methods sit between the above truly direct methods and dynamic sampling methods discussed below. They produce exactly independent samples with the exact probability density specified, but the number of steps needed to do this cannot be exactly predicted in advance. There is considerable freedom in designing a rejection algorithm to sample a given density, ρ.

The Monte Carlo practitioner occasionally must spend some time and (not that pleasant) effort optimizing parameters or otherwise tinkering to get a rejection method that is reasonably efficient. Rejection methods are often in the innermost loop of a Monte Carlo code, so their efficiency determines the running time of the code as a whole.

To generate a random variable with density $\rho(x)$, the rejection method uses independent random variables sampled from an auxiliary density, $\rho_0(x)$ and a "acceptance probability", $p(x)$. The method has two steps

Trial: generate a "trial" random variable, $X \sim \rho_0$. All trials are independent.

Rejection: "accept" the trial with probability $p(X)$. If X is accepted, it is the random variable generated by the algorithm. If X is rejected, go back to the trial step, generate a new (independent) X.

Accepting with probability p is done on the computer by comparing p to another (independent) uniform random variable. If p is larger (an event with probability p), accept. This trial and rejection process is repeated until a random variable is accepted. If ζ is the probability of getting an acceptance on any given trial, then the expected number of trials needed to get an acceptance is $1/\zeta$.

The following ghastly one line code C does all this:

```
while( unif() > acc_prob( X = trial()) );
```

This assumes that `float unif()` when called, returns a uniformly distributed random number, that `float acc_prob( float x)` returns the acceptance probability for trial variable `x`, and that `float trial()` returns a sample from the trial density, ρ_0. A code that works from lists could look like this:

```
while( unif[u_count++] > acc_prob( X = trial[t_count++])
   ) {
   if ( u_count >= U_LIST_SIZE ) unif_refil();
   if ( t_count >= T_LIST_SIZE ) trial_refil();
   }
```

In the second code fragment, `float unif[U_LIST_SIZE]` and `float trial[U_LIST_SIZE]` are arrays rather than subroutines. the procedures `void unif_refil()` and `void unif_refil()` refill the lists, using random number generators.

We can determine the probability density function for the eventual accepted X using the laws of conditional probability. It is given by

$$\begin{aligned}
\rho(x)dx &= \text{Prob}\,[\text{ accepted } X \in (x, x+dx)] \\
&= \text{Prob}\,[\text{ trial } X \in (x, x+dx) \mid \text{accepted } X] \\
&= \frac{\text{Prob}\,[\text{ trial } X \in (x, x+dx) \text{ and accepted } X\ ]}{\text{Prob}\,[\text{ got an acceptance }]} \\
&= \frac{1}{\zeta}\rho_0(x)dx \cdot p(x) \quad ,
\end{aligned}$$

where ζ is the probability of getting an acceptance on a given trial, as above. Putting this together gives

$$p(x) = \zeta \frac{\rho(x)}{\rho_0(x)} \quad . \tag{11}$$

In order for $p(x)$ as given in (10) to be a probability, it must be between 0 and 1. In order for this to be possible (with a fixed ζ), we must have

$$\rho(x) \leq \frac{1}{\zeta}\rho_0(x) \quad .$$

This requirement limits the possibilities for ρ_0. For example, one can sample a standard normal by rejection from an exponential (the ratio $e^{-x^2/2}/e^{-x}$ is bounded), but one cannot sample an exponential by rejection from a gaussian (the ratio $e^{-x}/e^{-x^2/2}$ is not).

The rejection algorithm has the advantage that it can be applied even when the probability densities are known only up to a multiplicative constant. This situation arises, for instance, whenever the Gibbs Boltzmann distribution (the "canonical ensemble") is used. In that notation, suppose $\phi(x)$ and $\phi_0(x)$ are two energy functions with corresponding probability densities

$$\rho(x) = \frac{1}{Z}e^{-\phi(x)} \quad , \quad \text{and} \quad \rho_0(x) = \frac{1}{Z_0}e^{-\phi_0(x)} \quad .$$

If we write the acceptance probability also in exponential form:

$$p(x) = e^{-\psi(x)} \quad ,$$

then the discussion leading to (10) now gives the formula

$$\psi(x) = \phi(x) - \phi_0(x) + \alpha \ ,$$

where

$$\alpha = \log(Z) - \log(Z_0) - \log(\zeta) \tag{12}$$

should be taken as small as possible, subject to the constraint that $\psi(x) \geq 0$ for all x (so the $p(x)$ is a probability). If we work with the functions, ϕ, ϕ_0, and ψ, then only α need ever be known, not Z, Z_0, or ζ.

Example: Let us consider the problem of generating a standard normal by rejection from an exponential. To begin with, we will sample from the exponential density with mean one and do rejection to get a random variable whose density function is the positive half of the gaussian, that is

$$\rho(x) = \begin{cases} \frac{2}{\sqrt{2\pi}} e^{-x^2/2} & \text{if } x > 0, \\ 0 & \text{otherwise} \end{cases} , \text{ and } \quad \rho_0(x) = \begin{cases} e^{-x} & \text{if } x > 0, \\ 0 & \text{otherwise} \end{cases} .$$

From the treatment where the constants are supposed to be known we get acceptance probability

$$p(x) = \zeta \frac{\sqrt{2}}{\sqrt{\pi}} e^{x - x^2/2} \ .$$

The largest possible ζ that gives $p(x) \leq 1$ for all $x > 0$ is

$$\zeta = \frac{\sqrt{\pi}}{\sqrt{2}} e^{-1/2} = .7602 \ ,$$

which is also the efficiency of the rejection method: the probability of getting an acceptance on a given trial is 76%.

Example. We want to sample from the density

$$\rho(x) = \frac{1}{Z} e^{x^4/4}$$

by rejection from a gaussian. At first we consider rejection from a standard normal. In that case,

$$\psi(x) = \frac{x^4}{4} - \frac{x^2}{2} + \alpha \ . \tag{13}$$

To make sure $\psi \geq 0$, we compute that the minimum of ψ is taken at $x = \pm 1$ ($\psi' = 0 \Rightarrow x^3 = x$). Thus, if $\alpha = -1/4$, then $min_x \psi(x) = 0$, as needed.

We try to improve the efficience of this rejection method by rejecting from a normal with variance $\sigma^2 \neq 1$. This makes $\phi_0(x) = x_2/x\sigma^2$, so (12) becomes

$$\psi(x) = \frac{x^4}{4} - \frac{x^2}{2\sigma^2} + \alpha \ ,$$

which now is minimized at $x = \pm 1/\sigma$, so $\alpha = -1/4\sigma^4$. To optimize the acceptance probability, ζ, by the best choice of σ. we use $Z_0 = \sqrt{2\pi\sigma^2}$, so , from (11),

$$\log(\zeta) = \log(Z) + \frac{1}{4\sigma^4} - \log(\sigma) + \frac{1}{2}\log(2\pi) \ .$$

To maximize this expression, we differentiate with respect to σ and set the derivative to zero. Since Z and 2π are independent of σ, this gives $\sigma_{\text{opt}} = 1$: our standard normal rejection was already optimal.