Sampling random variables and the Monte Carlo method

Anirban Bhattacharya

http://www.stat.tamu.edu/~anirbanb

January 18, 2018

1 Sampling from distributions

1.1 The Monte Carlo method

One of major take-home messages from this course is this: if we can draw (independent) samples from a probability distribution (with density f with respect to some dominating measure), we can essentially perform "any" computation with that distribution. Two points to note here before we begin. I placed independent in braces because we shall encounter situations where drawing independent samples is too complicated. Second, by "any" computation, I specifically mean calculating integrals (or equivalently, expectations) of the form

$$I = \int h(x)f(x)\,dx,$$

where h is some measurable function. If f has a complicated form which is beyond analytic tractability, then computing I may be a non-trivial endeavor. Why do we care about such integrals? Because, many quantities of interest regarding f can be expressed in the above form. For example, if $h(x) = x$, we recover the mean, $h(x) = x^2$ recovers the second moment, $h(x) = \mathbb{1}_{(-\infty,t)}(x)$ recovers the cdf at t, and so on.

Let us initially assume we have N independent samples $X_1, \ldots, X_N$ distributed as f. Then, by the strong law of large numbers,

$$\frac{1}{N}\sum_{g=1}^{N} h(X_g) \to \int h(x)f(x)\,dx, \text{ almost surely.}$$

Hence, for sufficiently large N, we can approximate the integral by the sum

$$\hat{I} = \frac{1}{N}\sum_{g=1}^{N} h(X_g) \approx \int h(x)f(x)\,dx.$$

This way of approximating integrals is called the *Monte Carlo* method. The average error is $\mathcal{O}(N^{-1/2})$, since

$$\mathbb{E}\left|\frac{1}{N}\sum_{i=1}^{N} h(X_i) - \int h(x)f(x)\,dx\right| = \mathcal{O}\left(\frac{1}{\sqrt{N}}\right).$$

Verify this! You may assume $\int h^2(x)f(x)\,dx$ is finite.

There is more to Monte Carlo than it immediately meets the eye. Suppose I want to approximate the cdf of a distribution (with density f) at a bunch of different points. Do we need collect a different set of random samples from f for each point? Or could we save a lot of effort by just

collecting one set of random samples and using it for every point? Deep results from probability theory, sometimes called uniform LLNs, guarantee exactly that. We won't have the scope to go deeper into this as this requires a good amount of empirical process theory. However, we can appreciate the strength of such results from the discussion below. A uniform LLN states something like

$$\mathbb{E} \sup_{g \in \mathcal{G}} \left| \frac{1}{N} \sum_{i=1}^{N} g(X_i) - \int g(x) f(x) dx \right| \leq \frac{CL}{\sqrt{N}},$$

where $\mathcal{G}$ is a class of "smooth" functions[1], C is a universal constant, and L is some property of the function class $\mathcal{G}$; see, e.g., the footnote. It is crucial to note that the supremum is inside the expectation, which, by Markov's inequality, implies that the sample $\{X_i\}_{i=1}^N$ is "good" to approximate the expectation of any function $g(X)$ with $g \in \mathcal{G}$. We get the same scaling of the approximation error with N as we get for one function.

Importance sampling Suppose I want to calculate $p = P(X > 4)$, where $X \sim N(0, 1)$, using Monte Carlo. We draw, say 50 independent samples from $N(0, 1)$, and set

$$\hat{p} = 50^{-1} \sum_{i=1}^{50} \mathbb{1}(X_i > 4).$$

Do you see a problem with this? Since the normal distribution has light tails, it is possible that none of these 50 samples exceed 4, and we get $\hat{p} = 0$. This is an extreme example, but this situation is often encountered more generally when the function h takes on larger values in the tails of f. A fix-up to this is the *importance sampling* method, where instead of drawing samples from f, we draw those from an *importance density* g. As a result, we need to include a correction factor in the Monte Carlo sum.

Algorithm.

(i) Draw $X_1, \ldots, X_N \sim g$ independently.

(ii) Set

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} \frac{h(X_i)\, f(X_i)}{g(X_i)}.$$

Note that, as before, $E(\hat{I}) = I$. We have assumed that the importance density g is positive everywhere, so that $\hat{I}$ is well-defined. This can be somewhat relaxed, we only need g non-zero when hf is non-zero. A general guideline for the choice of g, at least in the univariate setting, is to use heavy-tailed distributions, like the Cauchy or some other t distribution. Observe that $\text{var}(\hat{I}) = \sigma_I^2/N$, where

$$\sigma_I^2 = \int \left(\frac{h(x) f(x)}{g(x)} \right)^2 g(x) dx - I^2 = \int \frac{\{h(x) f(x) - I g(x)\}^2}{g(x)} dx.$$

Thus, a good importance density should aim to make $\int (hf)^2/g$ small. We can also write this quantity as $\int h^2(x) w^2(x) g(x) dx$, where $w(x) = f(x)/g(x)$ is the likelihood ratio, the ratio between the target density and the importance density.

A basic assumption we have made is that the likelihood ratio $w(x)$ can be evaluated at all points. However, a density may only be specified upto constants, and it may ot may not be easy to obtain the constant.

[1] for example, the class of Lipschitz functions on $[0, 1]$ with Lipschitz constant bounded by a positive number L, i.e., $|g(x) - g(y)| \leq L|x - y|$ for all x, y.

Example. Find $P(X > 4)$, where $X \sim f$ with $f(x) \propto e^{-|x|^{1.2}}, x \in \mathbb{R}$.

There is a trick, called self-normalizing importance sampling, which avoids explicit calculation of the constant. To describe that, suppose, in general, $f(x) = f_U(x)/C$, where C is an unknown constant, and f_U can be evaluated everywhere. Let $X_1, \ldots, X_N$ be i.i.d. g as before. Define, $w_U(x) = f_U(x)/g(x)$, and let

$$\hat{I}_{SN} = \frac{\sum_{i=1}^N h(X_i) w_U(X_i)}{\sum_{i=1}^N w_U(X_i)}.$$

Then, it can be shown that $\hat{I}_{SN} \to I$ almost surely (VFY!).

1.2 Basic sampling methods

The important question that now needs addressing is how do we get those independent samples from f. We shall encounter situations when f is a multivariate distribution with complicated dependency structures, and it may not be obvious at all how to draw samples from f. As a starting point, one can begin with univariate distributions. Even then, this is an enormous topic with entire books devoted to it; Luke Devroye's *"Non-uniform random variate generation"* being one prominent example. The good news is that there are now standard ways to draw from most common univariate distributions, with implementations available in standard software. Nevertheless, it is good to know some of the basic ideas here so that we don't have to spend hours on the web searching for code when we have a slightly nonstandard distribution. Throughout, we shall assume we have independent samples from $U(0,1)$ available. This is also a topic on its own; often called pseudo-random number generation, which we won't get into any details.

Inverse-cdf method (continuous case). Suppose f is absolutely continuous with cdf F. Draw $U \sim U(0,1)$ and set $X = F^{-1}(U)$. Then, $X \sim f$.

The inverse-cdf method can be readily used to sample from random variables whose inverse-cdf has a nice closed form. For example, if (i)$X \sim$ Expo(1), then $F_X(x) = 1 - e^{-x}$ for $x > 0$ and 0 otherwise, (ii) If $X \sim$ standard-Cauchy, then $F_X(x) = 1/2 + (1/\pi)\arctan(x)$. Then, verify that, if $U \sim U(0,1)$, then (i) $-\log(1-U) \sim$ Expo(1), (ii) $\tan(\pi(U - 1/2)) \sim$ standard Cauchy.

Qn: The standard double-exponential or Laplace distribution has density $f(x) = (1/2)e^{-|x|}$ for $x \in \mathbb{R}$. How will you sample from a Laplace distribution?

Let's see a slightly non-trivial application of the inverse cdf method. Suppose we want to draw from a truncated distribution with density proportional to $f\mathbb{1}_{(u,v)}$, where f itself is a density on the real line, with cdf F. Suppose f is a standard distribution which our software can sample from (e.g., normal, gamma,...). Consider the following algorithm: draw $U \sim U(F(u), F(v))$ and set $X = F^{-1}(U)$. Then, $X \sim f\mathbb{1}_{(u,v)}$. (VFY!)

Although simple to describe, the main limitations of the inverse cdf method are (i) F may not be analytically tractable, and even if analytically tractable, may not be easily invertible, and (ii) mainly limited to univariate distributions.

Inverse-cdf method (discrete case). We shall often use the shorthand $X \sim \text{Cat}(K; \pi_1, \ldots, \pi_K)$ to denote a discrete distribution supported on $\{1, \ldots, K\}$, with $Pr(X = j) = \pi_j$. Note: this is the same as the multinomial distribution with K boxes and one trial.

Define the cumulative sums $c_j = \sum_{i=1}^j \pi_i$ for $j = 1, \ldots, K$. Note that $c_K = 1$. Also, define $c_0 = 0$. Draw $U \sim U(0,1)$. Find $j^* \in \{1, \ldots, K\}$ such that $U \in (c_{j^*-1}, c_{j^*})$. Set $X = j^*$. Then, verify that $X \sim \text{Cat}(K; \pi_1, \ldots, \pi_K)$.

Mixture distributions. Suppose $f_1, \ldots f_K$ are densities and $\pi_1, \ldots, \pi_K$ be non-negative

weights adding up to one. The mixture density

$$f(x) = \sum_{h=1}^{K} \pi_h f_h(x).$$

Suppose I can draw samples from any of the densities f_hs. How would draw a sample from the mixture? Here is a hint: the mixture density can be hierarchically (again!) represented as follows (VFY!).

$$X \mid Z = h \sim f_h, \quad Z \sim \mathrm{Cat}(K; \pi_1, \ldots, \pi_K).$$

Hierarchically specified distributions. More generally, suppose $f(x) = \int g(x \mid z)p(z)dz$, where p is a density, and for any z in the support of p, $g(\cdot \mid z)$ is also a density. To sample from f, first draw $z \sim p$, and then draw $x \sim g(\cdot \mid z)$. Then, $x \sim f$. (VFY!)

This is a simple fact, but extremely useful. For example, if we can sample from normal and gamma distributions, the above automatically suggests a sampling mechanism for any t distribution. (WHY?)

Sampling from MVN Standard softwares now have samples from MVN implemented, but in many situations it may be computationally expensive to call the built-in function directly.

Suppose we want to sample from $\mathcal{N}_d(\mu, \Sigma)$.

1. Perform a (lower) Cholesky decomposition $\Sigma = LL'$.
2. Set $Z = (Z_1, \ldots, Z_d)$, where Z_is are iid $N(0, 1)$.
3. Set $X = \mu + LZ$.

Then, $X \sim \mathcal{N}_d(\mu, \Sigma)$.

Now suppose a situation arises where we want to sample from $\mathcal{N}_d(\mu_i, \sigma_i^2\Sigma)$ for $i = 1, \ldots, T$, where T is some large number (say, in the order of tens of thousands), and d is in the order of hundreds. How shall you go about it?

Another situation that often arises is when the MVN we want to sample from is in the form,

$$f(x) \propto e^{-\frac{1}{2}(x'Qx - 2b'x)}, \quad x \in \mathbb{R}^d,$$

which we know is the $\mathcal{N}(Q^{-1}b, Q^{-1})$ distribution. If we wanted to feed this directly into software, we would need the covariance matrix, for which we need to invert Q, an expensive and potentially unstable operation. The following sampling algorithm (Rue, 2001) avoids calculating the inverse of Q and only requires a Cholesky factorization and a series of linear system solutions, both of which are more efficient and stable compared to computing inverse.

- Perform a Cholesky decomposition $Q = LL^{\mathrm{T}}$, where L is lower triangular.
- Draw $z \sim N(0, \mathrm{I}_d)$, solve $L^{\mathrm{T}}y = z$.
- Solve $L^{\mathrm{T}}\theta = v$, where $Lv = b$.
- Set $X = y + \theta$.

Then, X produced as above has the desired $N(Q^{-1}b, Q^{-1})$ distribution (VFY!).

1.3 Rejection sampling

Suppose we want to sample from

$$f(x_1, x_2) \propto e^{-\frac{1}{2}(x_1^2+x_2^2)}, \quad x_1^2 + x_2^2 \leq 1.$$

You can recognize this as the bivariate standard Gaussian distribution truncated to the unit disk. How shall we go about it?

One idea is to keep sampling from the bivariate standard Gaussian distribution until we get a sample which falls inside the disk. This is the core idea of rejection sampling. The following result justifies why this is valid.

Theorem. *Let f be a density (w.r.t. Lebesgue measure) on $\mathbb{R}^d$ and let $A \subset \mathbb{R}^d$ be a Borel set with $\int_A f(u)du = p > 0$. Let $g \propto f\mathbb{1}_A$ be the density of f truncated to A.*

Let $X_1, X_2, \ldots$ be i.i.d. draws from f and let Y be the first X_i taking value in A. Then, $Y \sim g$.

Proof. We have, for any Borel set $B \subset \mathbb{R}^d$,

$$\begin{aligned} P(Y \in B) &= \sum_{j=1}^{\infty} P(X_1 \notin A, \ldots, X_{j-1} \notin A, X_j \in B \cap A) \\ &= \sum_{j=1}^{\infty} (1-p)^{j-1} P(X_1 \in B \cap A) \\ &= \frac{P(X_1 \in B \cap A)}{P(X_1 \in A)} \\ &= \int_B g(u)du. \end{aligned}$$

We now present the rejection sampling algorithm in a more general and most widely known form. Before that, we need another fun fact.

Theorem. *Let f be a density on $\mathbb{R}^d$, and define $G = \{(x, u) \in \mathbb{R}^{d+1} : 0 \leq u \leq cf(x)\}$ for any positive constant c.*[2]

(i) If (X, V) is a random variable in $\mathbb{R}^{d+1}$ uniformly distributed on G, then $X \sim f$.

(ii) If $X \sim f$ and $U \sim U(0, 1)$ are independent, then $(X, cU\, f(X))$ is uniformly distributed on G.

Proof. Let's prove (i) first. Observe that, for any Borel set $B \subset \mathbb{R}^d$,

$$P(X \in B) = P[(X, V) \in B_1],$$

where $B_1 = \{(x, u) \in \mathbb{R}^{d+1} : x \in B, 0 \leq u \leq cf(x)\}$. Now,

$$P[(X, V) \in B_1] = \frac{\text{area}(B_1)}{\text{area}(G)} = \frac{c\int_B f(u)du}{c} = \int_B f(u)du,$$

where the first equality follows since (X, V) is uniform on G, and the subsequent one simply uses that area under a curve is the integral of the function. Thus, we have proved that for any

[2] Observe that G is the area between the x-axis and the graph of cf

Borel B,

$$P(X \in B) = \int_B f(u)du,$$

implying $X \sim f$.

(ii) Exercise!

Part (i) of the theorem relays an important fact. If we can sample uniformly from the area under (any constant times) a density, then that automatically leads to a sample from that density. While part (ii) tells us one way to perform such a uniform sampling, it is not immediately useful because it in itself requires a sample from the target density, which we are trying to achieve in the first place. This is where rejection sampling comes into play, where we use a different density to draw the X sample from, and then accept that sample under some condition.

Algorithm: Rejection Sampling

Let g be a density on $\mathbb{R}^d$, which is easy to sample from, and satisfies $f(x) \leq Mg(x)$ for all x (in the support of f), for some constant $M \geq 1$.

Step 1. Sample $X \sim g$ and $U \sim U(0,1)$ independently.

Step 2. If $U \leq f(X)/\{Mg(X)\}$, accept X as a sample from f. Otherwise, return to Step 1.

The intuitive explanation is as follows. Remember, our goal is to uniformly sample from the area under f. To do so, we sample uniformly from a larger area, namely the area under Mg, and keep drawing such samples until we get one in the desired region. A combination of the two theorems delivers the desired result. (VFY!)

One comment here is that we once again encounter the likelihood ratio $f(x)/g(x)$, as we did in importance sampling.

Even before we investigate the algorithm any further, it should be evident that the efficiency of the algorithm should depend on how big M is. Choosing M very large, it is easier to satisfy the inequality $f(x) \leq Mg(x)$ for all x. On the other hand, the ratio of the area under f and Mg is $1/M$, which means on an average we shall need M draws from g to get one draw from f. This can be made concrete quite easily. Let N be the number of (X, U) pairs drawn before the algorithm halts. Then, it can be shown that $N \sim \text{Geometric}(1/M)$, and thus $E(N) = M$(VFY!).

Example. Suppose we want to sample from the standard normal distribution. We have seen previously that it is easy to directly sample from the Cauchy distribution using the inverse cdf method. If $f(x) = (2\pi)^{-1/2}e^{x^2/2}$ is the standard normal distribution, and $g(x) = \pi^{-1}(1+x^2)^{-1}$ is the standard Cauchy distribution, then we need to find an M such that $f(x) \leq Mg(x)$ for all x. We can simply set

$$M = \sup_x \frac{f(x)}{g(x)} = \sup_x \sqrt{\pi/2}(1+x^2)e^{-x^2/2},$$

which is finite and equals 1.52. Thus, the acceptance rate of the rejection sampler will be $(1/1.52) \times 100\% = 65.8\%$.

Qn. As in importance sampling, we need to be able to evaluate the likelihood ratio $f(x)/g(x)$ to run the above version of the rejection sampler. Suppose we do not know the normalizing constant of f, that is $f(x) = \widetilde{f}(x)/C$, where $\widetilde{f}$ is a function we can evaluate and C is unknown. Can you modify the rejection sampler without using the value of C?