Sampling - Part 1

Kevin Shi

May 2025

Contents

1 Introduction

Over the course of implementing statistical inference methods, I will need access to some data to test if my code works or not, and to make sure my understanding is correct. Being somewhat lazy, I would like to generate my own numerical data rather than find some dataset online.

Unfortunately, I am not quite lazy enough, and I have decided to implement some cool sampling algorithms on my own, rather than use pre-existing samplers. Therefore, I have resolved to go over several sampling methods theoretically here, before implementing them in code.

A key assumption will be that we have access to some method of sampling uniformly from the interval $[0, 1]$, which is in the realm of computer science. Something else to clarify here is that I mean sampling in the sense that I want to draw a sample from a distribution, not necessarily that I want to be physically sampling from a population.

The textbooks I used for this are *Non-Uniform Random Variate Generation* by Luc Devroye, and *Monte Carlo Statistical Methods* by Robert and Casella.

2 Basic Sampling Methods

2.1 Inverse CDF Sampling

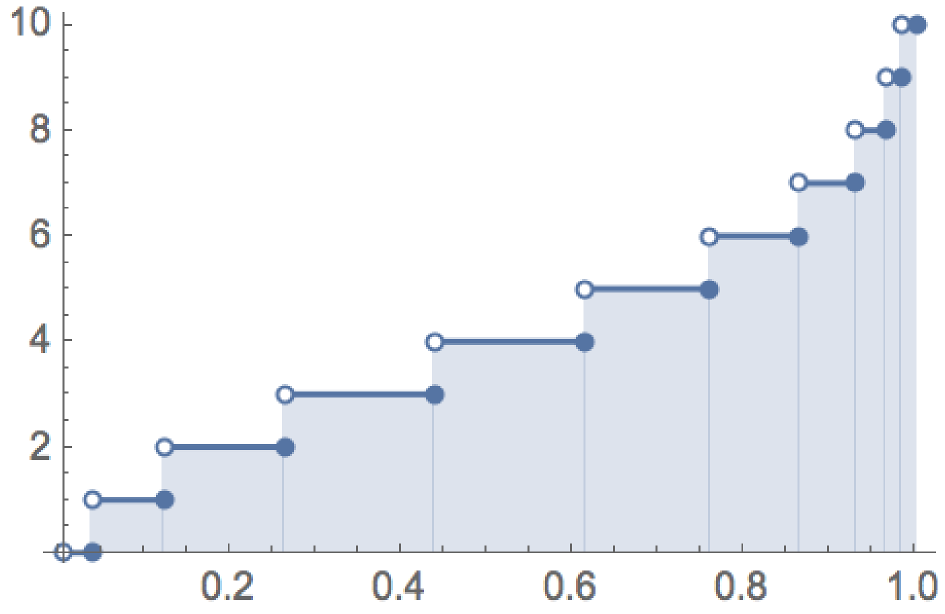
An elegant way of sampling from a distribution analytically is by the inverse CDF method. Say we have a distribution whose CDF is $F(x)$, for $x \in X$, $X \subseteq \mathbb{R}$. We are given a sample u drawn from a standard uniform distribution. The most general form of this method states that:

$$F^{-1}(u) = \inf\{x; F(x) \geq u\} \tag{1}$$

Note the condition does not require F to be invertible, as it simply defines the a new function F^{-1}. The reasoning is that we would like to generalize this method to discrete random variables.

For continuous random variables, the set of $\{x; F(x) \geq u\}$ contains exactly one element x, such that the infimum of this set is trivially x, and $x = F^{-1}(u)$.

For discrete random variables, given the CDF $F(x)$, we are essentially just drawing out a step function where the steps occur as soon as the uniform variable takes a value $u > F(x)$. In some sense we are 'binning' amounts of uniform 'probability mass' to match the probability mass of the discrete variable X. There is a nice image I found here that shows what this looks like (for a Poisson random variable):
To verify that this is correct, we start with the CDF of X:



$$F(x) = P(X \leq x) = P(F^{-1}(U) \leq x) = P(\inf\{y; F(y) = U\} \leq x) \tag{2}$$
$$= P(U \leq F(x)) = F(x) \tag{3}$$

It is also a fact that applying the distribution function ***to*** the random variable X (as a transformation) results in a standard uniform variable.

$$P(F(X) \leq u) = P(X \leq F^{-1}(u)) = F(F^{-1}(u)) = u \tag{4}$$

Although in my view this is the most desirable way to generate samples from a distribution as it produces analytic, exact solutions, for complicated distributions the inverse must be computed using numerical solutions. Furthermore, there are complications with multidimensional/vector-valued random variables, where the inverse is not well defined. For example, consider a 2-dimensional multivariate Gaussian with identity covariance matrix. This implies that we can factor the density into two univariate Gaussians. If we wanted to compute the CDF as some vector/point $[z, -z]$, we have

$$F_{XY}(z, -z) = \int_x \int_y f_{XY}(z, -z) dy dx$$

But we can clearly see through this formulation that a separate point $[-z, z]$ also produces the same CDF, meaning the CDF is not one-to-one (and therefore not bijective), and so it is not invertible.

(Note: There is a generalization where this concept can be extended to multiple dimensions, but it must be modified. See here for more details).

Therefore, I will focus most of my attention on Monte Carlo methods, which when properly implemented can be quite robust in their own right, and is the standard method of sampling today. For example, the numpy sampler for the normal distribution uses the Ziggurat algorithm, a rejection sampling method that at its core is a Monte Carlo method.

3 Monte Carlo Sampling

One of the most fundamental ideas that simple Monte Carlo methods is to obtain quantities of interest through simulation rather than through analytical or standard numerical methods. For example, we might want to calculate the ratio of an inscribed circle to a unit square, which we can do by uniformly generating coordinates (x, y) inside the square and calculating the ratio of the number of points inside the circle to the total number of points.

In our case, we wish to use Monte Carlo methods to generate samples from a particular distribution, which is a different task than the integration task I gave as an example earlier.

3.1 Rejection Sampling

One of the main ideas underpinning rejection sampling is the following theorem from chapter 2 of Devroye:

Theorem 1 (*Simulation Theorem*)**.** Let X be a random vector with density f on $\mathbb{R}^d$, and let U be an independent standard uniform random variable. Then the joint $(X, cUf(X))$ is uniformly distributed on $A = \{(x, u) : x \in \mathbb{R}^d, 0 \leq u \leq f(x)\}$, where $c > 0$ is some arbitrary constant. Vice versa, if (X, U) is a random vector in $\mathbb{R}^{d+1}$ uniformly distributed on A, then X has density f on $\mathbb{R}^d$.

This might seem somewhat abstract, but let's try to think about what this is saying. For our purposes, don't worry about the first part. Only the second part is immediately important for implementing rejection sampling. Essentially this is saying that if we sample $(X, U) \sim \mathcal{U}(\{(x, u) : x \in \mathbb{R}^d, 0 \leq u \leq f(x)\})$, then $X \sim f(x)$, meaning that we have equivalently sampled from $f(X)$!

Of course, the question is how do we sample the vector (X, U). We might want to sample $X \sim f(x)$ and then $U|X = x \sim \mathcal{U}(0, f(x))$, but this is completely pointless as the entire purpose of sampling (X, U) was to generate samples from $f(x)$ in the first place. Alternatively we could try to sample $U \sim \mathcal{U}(0, 1)$ and then sample $X|U$, but the conditional is usually not computable.

Therefore the idea is to sample points (Y, U) from a larger set that completely encloses the original set A, and then only count it if it satisfies the conditions in set A.

Consider as an example the case where $f(x)$ has support between $[a, b]$, and that the density is bounded by some constant m. Then for the pair $(Y, U) \sim (0 \leq u \leq m)$ we first sample $Y \sim \mathcal{U}(a, b)$ (sampling from the larger enclosing set), and then sample $U|Y = y \sim \mathcal{U}(0, m)$, only taking the pair if the condition $0 \leq u \leq f(y)$ is met. If we consider the distribution of the accepted random variable as X, then we have the correct distribution, as:

$$P(X \leq x) = P(Y \leq x | U \leq f(Y)) = \frac{P(Y \leq x, U \leq f(Y))}{P(U \leq f(Y))} \tag{5}$$

$$= \frac{\int_a^x \int_0^{f(y)} du dy}{\int_a^b \int_0^{f(y)} du dy} = \frac{\int_a^x f(y) dy}{\int_a^b f(y) dy} \tag{6}$$

$$= \int_a^x f(y) dy \tag{7}$$

Where we obtain the result through Bayes' rule followed by the law of total probability. This means that if we sample (Y, U) from the larger set and then only keep those samples (defined as random variable X) that satisfy the stricter condition that $0 \leq u \leq f(y)$, then we are sampling uniformly from the distribution $f(x)$, meaning that the distribution of X is indeed the one we are after.

This can be extended to non-uniform 'enclosing' distributions, so long as the larger function is always greater than the desired density $f(x)$ on the support of the distribution. It also must be possible to sample uniformly from this larger set defined by points bounded by $cg(x)$, where $c > 1$ is some constant. As $cg(x)$ is not a distribution as it must integrate to a value greater than 1 (as it is larger than $f(x)$ at all points), we take $g(x)$ to be some arbitrary, valid distribution to allow uniform sampling (via inverse CDF or some alternative method).

The setup is much the same: sample $Y \sim g$, then sample $U|Y = y \sim U(0, cg(y))$, accepting the sample if $0 \leq u \leq f(y)$. We can verify is a similar way that this is valid, saying that A is a measurable set (for continuous RVs it might be any arbitrary interval):

$$P(X \in A) = P(Y \in A | U \leq f(Y)) = \frac{P(Y \in A, U \leq f(Y))}{P(U \leq f(Y))} \tag{8}$$

$$= \frac{\int_A \int_0^{f(y)} \frac{du}{cg(y)} g(y)dy}{\int_A \int_0^{f(y)} \frac{du}{cg(y)} g(y)dy} = \int_A f(y)dy \tag{9}$$

For every measurable set A. In the final algorithm (which I have taken from Robert and Casella's book), it is easier to sample U from a standard uniform and instead make the condition $U \leq \frac{f(X)}{cg(X)}$, which is equivalent to the earlier formulation.

To make it explicit, the general steps of the basic rejection sampling (or accept-reject) method are:

1. Identify $c > 1$ and $g(x)$ such that $cg(x) \geq f(x)$ everywhere.
2. Sample $X \sim g$
3. Sample $U \sim \mathcal{U}(0, 1)$
4. Accept $Y = X$ if $U \leq \frac{f(X)}{cg(X)}$
5. Otherwise, return to step 2.

For as many samples as we need, just start from step 2 until a sufficient number of samples have been generated.

In some sense, if we consider a univariate distribution that we wish to model, we can sort of conceptualize this as throwing darts onto a flat plane of the graph of the density, while making sure that the entire density 'fits' on the pages of this graph. As long as a dart lands underneath the graph line, it is a sample from the distribution. This algorithm also extends into higher dimensions.

Rejection sampling has nice intuition, but it has some major flaws that motivate more intelligent Monte Carlo sampling techniques, which I will go over in the next part.