# Stochastic Sampling

This section describes computational methods that can be used to draw random realisations, or samples from arbitrary probability distributions. These methods can be used to learn about any probability distribution, however our main goal is to use them to study the posterior distributions that arise when performing Bayesian inference.

## 0.1 Why Monte Carlo?

Given a *probability distribution* $P$ (sometimes called the *target distribution*) a set of *stochastic samples* are $n$ independent draws from this distribution;

$$x \overset{\text{iid}}{\sim} P, \quad \text{for } i = 0, 1, \ldots n-1. \tag{1}$$

For a large enough $n$, such stochastic samples can be used to answer virtually any question one might have about the distribution $P$. They are commonly used in statistical inference, especially Bayesian inference. They can be used for a variety of purposes; for example:

- approximating the full $d$-dimensional distribution – e.g. by generating a $d$-dimensional dimensional histogram or kernel density estimate (KDE) of the samples;
- approximating and/or visualising the marginal distributions – e.g. by generating a 1- or 2-dimensional histograms or KDEs of the marginal distributions of $P$ for use in a *corner plot*;
- computing *credible intervals* (or *credible regions*) – if $P$ is a posterior in a Bayesian analysis then stochastic samples can be used to quantify measurement uncertainties by finding intervals within which parameters fall with a particular probability;
- approximating integrals of the form $I = \int \mathrm{d}x \; \varphi(x)P(x)$ – this can be done via the Monte-Carlo sum $S_n = \frac{1}{n}\sum_{i=0}^{n-1} \varphi(x_i)$ which satisfies $S_n \to I$ as $n \to \infty$.

Note that in all the above applications the stochastic samples are being used to calculate a deterministic quantity. This is actually a reasonable definition of a *Monte-Carlo method.*

The final bullet point, Monte Carlo integration, is the most important application. It's worth emphasising the draw backs of Monte Carlo integration[1]. The error on the estimator $S_n$ for the integral $I$ scales as $\sim 1/\sqrt{n}$. This is actually *not* very impressive; in contrast, deterministic quadrature methods exist with *exponential* convergence with the number of integration nodes per dimension (at least for smooth integrands). However, in $d$ dimensions the number of integration nodes required increases exponentially with $d$. In contrast, Monte Carlo methods converge as $\sim 1/\sqrt{n}$ in any number of dimensions. From this point of view, Monte Carlo methods are bad, but for high-dimensional problems there may not be anything better.

## 0.2 Why is stochastic sampling hard?

We will review a few basic methods and see why they struggle in high dim problems.

- Transform sampling, Sec. 0.2.1
- Rejection sampling, Sec. 0.2.2

### 0.2.1 Transform Sampling

For some simple distributions a function can be found that maps from a known distribution, $Q$, to the target distribution, $P$.

Given a random variable $x \sim Q$ and an invertible transformation $f : \mathbb{R}^d \to \mathbb{R}^d$ the random variable $y = f(x)$ is distributed as $y \sim P$, where

$$P(y) = \left|\frac{\partial f}{\partial x}\right|^{-1} P(x). \tag{2}$$

The $d \times d$ matrix $\frac{\partial f}{\partial x}$ is the Jacobian of the transformation.

The challenge is to design a transformation $f$ such that $y$ follows the desired target distribution. Unfortunately, for an arbitrary target distribution in multiple dimensions, this is usually not possible to do excatly.

[1] See the nice discussion in Sokal (1997) "Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms", doi:10.1007/978-1-4899-0319-8_6.

In 1 dimension the problem can be solved using the inverse of the CDF.

The *transform sampling* method, or *inverse CDF sampling* method, starts with a uniform random variable $u \sim \mathcal{U}(0,1)$ and applies a specific transformation $x \equiv F^{-1}(u)$ designed such that the resulting random variable follows the target distribution, $x \sim P$.

The *cumulative distribution function* (CDF) $F(x)$ is defined in terms of the PDF via

$$F(x) = \int_{-\infty}^{x} \mathrm{d}x' \, P(x'). \tag{3}$$

Because $P$ is a non-negative, normalised PDF, the CDF is a monotonically increasing function satisfying $\lim_{x\to-\infty} F(x) = 0$ and $\lim_{x\to-\infty} F(x) = 1$.

If $F$ is continuous, then it has a unique inverse, $F^{-1}(F(x)) = x$. The function $F^{-1}$ is known variously as the *inverse CDF*, the *percent-point function* (PPF), or *quantile function.*

Given a sample $u \sim \mathcal{U}(0,1)$, the inverse CDF can be used to produce a sample $x \sim P$.

**Theorem 0.2.1.** *If $u \sim \mathcal{U}(0,1)$, then $x \equiv F^{-1}(u)$ satisfies $x \sim P$.*

*Proof.* Because $\mathcal{U}(0,1)$ has support in the interval (0,1) and $0 \leq F(x) \leq 1$, it follows that the distribution of $x$ has the same support as $P$. From its definition, the PDF of $x$ is $P_{\mathcal{U}}(u)|\mathrm{d}u/\mathrm{d}x|$. Differentiating Eq. 3 gives $\mathrm{d}u/\mathrm{d}x = P(x)$, and using the fact that $P(u) = 1$ in the supported region, shows that this equals $P(x)$, as required. □

Transform sampling using the inverse CDF sampling is illustrated in Fig. 1.

Note, inverse CDF sampling requires finding the CDF (by integrating the PDF) and inverting it; in general, both of these operations might have to be done numerically.

**Box 0.1: Using inverse CDF method to sample the standard normal (a.k.a. Gaussian) distribution**

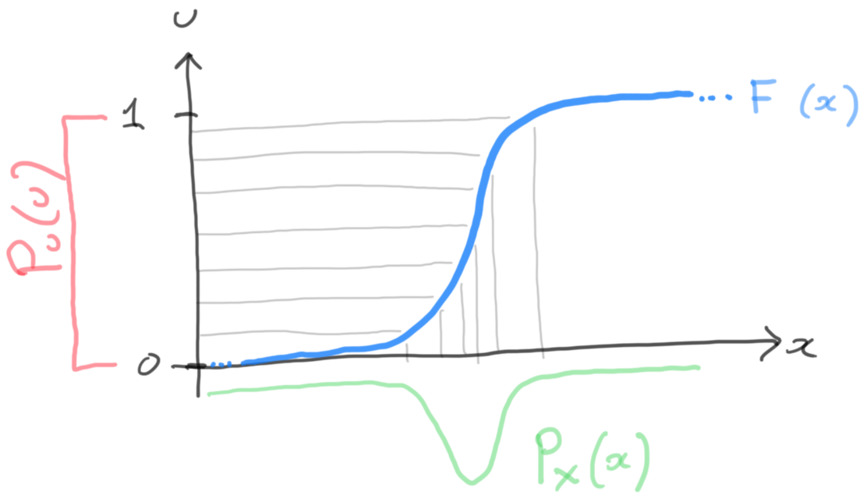


Figure 1: An illustration of the inverse CDF method for sampling from a univariate distribution. Applying the function $F^{-1}$ maps a uniformly distributed random variable to a new random variable that follows the desired distribution, $x \sim P$.

The target PDF is

$$P(x) = \frac{\exp\left(\frac{-1}{2}x^2\right)}{\sqrt{2\pi}}. \tag{i}$$

A random variable from this distribution is denoted $x \sim \mathcal{N}(0, 1)$.

The target CDF is

$$F(x) = \int_{-\infty}^{x} \mathrm{d}x' \; P(x') = \frac{1}{2}\left(1 + \mathrm{erf}\left[\frac{x}{\sqrt{2}}\right]\right), \tag{ii}$$

where the error function is defined as $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \mathrm{d}y \; \exp(-y^2)$.

The target inverse CDF is

$$F^{-1}(x) = \sqrt{2}\,\mathrm{erf}^{-1}(2x - 1), \tag{iii}$$

where the inverse error function satisfies $\mathrm{erf}^{-1}\big(\mathrm{erf}[x]\big) = x$.

Therefore, if $u \sim \mathcal{U}(0, 1)$, then $x = \sqrt{2}\,\mathrm{erf}^{-1}(2u - 1)$ is distributed as $x \sim \mathcal{N}(0, 1)$.

Samples from the normal distribution $\mathcal{N}(\mu, \sigma^2)$ can be obtained by applying the shift and scaling transformations $\mu + x\sigma$.

**Box 0.2: Using inverse CDF method to sample the exponential distribution**

The target distribution is the exponential distribution (with rate parameter $\lambda > 0$) which has PDF

$$P(x) = \begin{cases} \lambda \exp(-\lambda x) & \text{if } x > 0 \\ 0 & \text{else} \end{cases}. \tag{i}$$

A random variable from this distribution is denoted $x \sim \text{Exp}(\lambda)$.

The target CDF is

$$F(x) = \int_0^x \mathrm{d}x' \; P(x') = 1 - \exp(-\lambda x). \tag{ii}$$

The target inverse CDF is

$$F^{-1}(x) = \frac{-\log(1-x)}{\lambda}. \tag{iii}$$

Therefore, if $u \sim \mathcal{U}(0,1)$, then $x = -\log(1-u)/\lambda$ is distributed as required. However, $(1-u)$ is distributed identically to $u$ (this follows by symmetry), so we may instead use the simpler expression $x = -\log(u)/\lambda$.

### 0.2.2 Rejection Sampling

In one dimension, the *rejection sampling method* aims to sample a 1D random variable by first uniformly sampling in 2D and keeping only the first coordinate of points that satisfy a certain condition. It is also commonly called the *accept-reject algorithm* and is sometimes attributed to John von Neumann.

The rejection method uses a *proposal distribution* $Q$. We assume that we are already able to sample efficiently from $Q$ (e.g. by using the inverse CDF method described in 0.2.1). We define a constant $1 < M < \infty$ such that $P(x) \leq MQ(x)$ for all $x$. Note that this requires that $Q$ must have support everywhere that $P$ has support.

The rejection sampling algorithm proceeds as follows. Random samples are obtained from both the proposal $Q$ and uniform $\mathcal{U}$ distributions. (The ordered pair $(x, Mu)$ can be thought of as a point in two dimensions.) If a certain acceptance condition is met then $x$ is accepted as a sample from $P$, otherwise $x$ is rejected and we try again.

**Algorithm 0.1** Rejection Sampling

```
1: $x \sim Q$                              ▷ Sample the proposal
2: $u \sim \mathcal{U}(0,1)$
3: if $Mu < P(x)/Q(x)$ then
4:     return $x$                          ▷ Accept
5: else
6:     go to line 1                        ▷ Reject
7: end if
```

**Theorem 0.2.2.** *The random $x$ produced from Alg. 0.1 is distributed as $x \sim P$.*

*Proof.* It will suffice to show the PDF of the conditional distribution of $x$ given $Mu < P(x)/Q(x)$ produced by the algorithm is $P(x)$. Bayes' theorem gives

$$\text{Prob}\left(x = x \middle| u < \frac{P(x)}{MQ(x)}\right) = \text{Prob}\left(u < \frac{P(x)}{MQ(x)} \middle| x = x\right) \frac{\text{Prob}(x = x)}{\text{Prob}\left(u < \frac{P(x)}{MQ(x)}\right)}. \tag{4}$$

Using $x \sim Q$ and $u \sim \mathcal{U}(0,1)$, the factors on the right-hand side can be written

$$\text{Prob}\left(u < \frac{P(x)}{MQ(x)} \middle| x = x\right) = \frac{P(x)}{MQ(x)}, \tag{5}$$

$$\text{Prob}(x = x) = P(x), \tag{6}$$

$$\text{and}\ \ \text{Prob}\left(u < \frac{P(x)}{MQ(x)}\right) = \int_{-\infty}^{\infty} \mathrm{d}x\, \frac{P(x)}{MQ(x)} Q(x) = \frac{1}{M}. \tag{7}$$

Substituting Eqs. 5, 6 and 7 back into Eq. 4 gives the result,

$$\text{Prob}\left(x = x \middle| u < \frac{P(x)}{MQ(x)}\right) = P(x). \tag{8}$$

□

An illustrated of the rejection sampling method is shown in Fig. 2.

The rejection algorithm takes an average of $M$ iterations per sample. For best performance the proposal $Q$ should be as similar as possible to the target $P$ so that $M$ can be chosen to be as small as possible (thereby avoiding "wasted space"). Ideally, $Q$ should also be chosen such that it is easy to sample from efficiently.

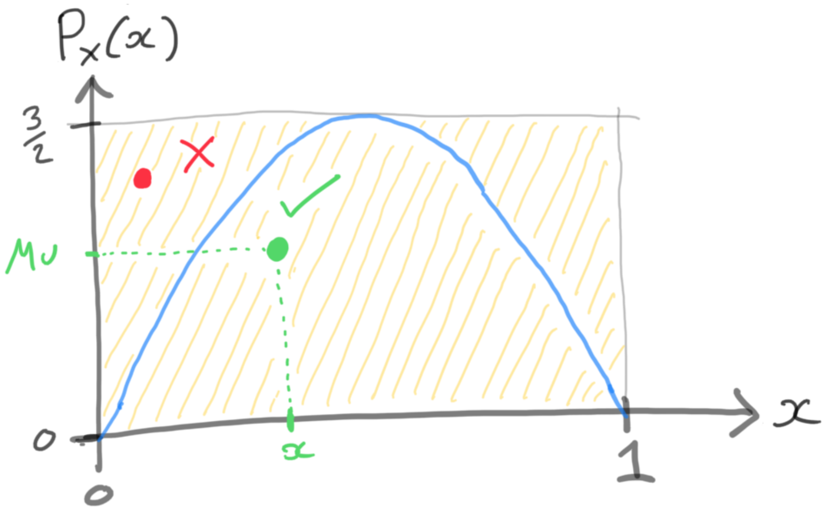


Figure 2: An illustration of rejection sampling for the beta distribution $x \sim \text{Beta}(\alpha = 2, \beta = 2)$ with $P(x) = 6x(1-x)$ (blue curve). The proposal distribution is chosen to be the uniform distribution, $Q = \mathcal{U}(0,1)$. The smallest possible choice for the constant is $M = 3/2$ which corresponds to drawing points uniformly in the yellow shaded region and keeping/rejecting the $x$-coordinates of the green/red points that lie below/above the line.

In practice, the difficult part of implementing rejection sampling is finding a proposal distribution that is (i) easy to sample from and (ii) encompasses the target distribution without lots of wasted space allowing us to find as small an $M$ as possible. Choosing a good proposal distribution is not always easy (see, for example, example 0.3). To circumvent this, several variations of *adaptive rejection sampling* algorithms exists that aim to build a self-tuning proposal distribution that updates and improves itself each time a sample is rejected. These adaptive methods aim to improve the sampling efficiency in the limit of a large number of samples and remove the burden of choosing the proposal distribution from the user. These adaptive variations of the method are not discussed in detail here.

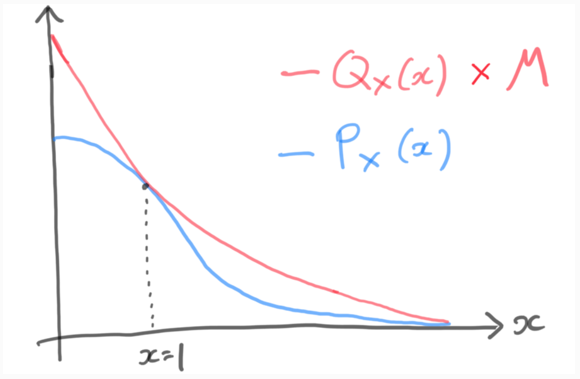
Rejection sampling can also be used for multivariate distributions. (Nothing in the above algorithm required $P$ to be univariate.) However, in high dimensions the condition $P(x) \leq MQ(x)$ usually forces $M$ to be so large that the method becomes inefficient, unless $Q$ is an extremely good approximation to $P$. In fact, even finding a suitable $M$ can be difficult because we do not generally know in advance where the peaks of $P(x)$ are located or how tall they are. For these reasons, rejection sampling is usually only used for univariate distributions.

**Box 0.3: Using the rejection method to sample the standard normal distribution**

We have already seen how to sample the normal distribution with the inverse CDF method (Example 0.3) now we see how to do it using the rejection method.

We chose as our proposal $Q$ the exponential distribution with unit rate parameter. We already know how to sample from $Q$ (see Example 0.2). Because $Q$ only has support for $x < 0$, we use the rejection method to sample from the normal distribution truncated to $x > 0$ and then make a random choice for the sign.

The ratio of the PDF of our target distribution to the PDF of our proposal distribution is $\exp(x - x^2/2)/\sqrt{2\pi}$. This ratio achieves its maximum of $\sqrt{2e/\pi}$ when $x = 1$. Therefore, we choose $M = \sqrt{2e/\pi}$. (See figure below.)



*Step i:* Generate two independent uniform random variables; $u_1, u_2 \overset{\text{iid}}{\sim} \mathcal{U}(0,1)$.

*Step ii:* If $u_2 < \exp\left(\frac{-(1-x)^2}{2}\right)$ then let $|x| = -\log u_1$, else go back to *Step i*.

*Step iii:* Choose the sign. E.g. by generating another uniform random variable $u_3 \sim \mathcal{U}(0,1)$ and if $u_3 < 1/2$, then let $x = |x|$, else let $x = -|x|$.

The resulting $x \sim \mathcal{N}(0,1)$, as required.