Sampling from Probability Distributions

Anthony Charles

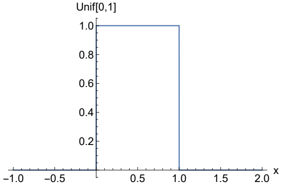
July 6, 2023

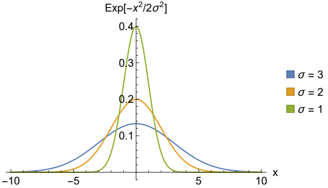
Outline

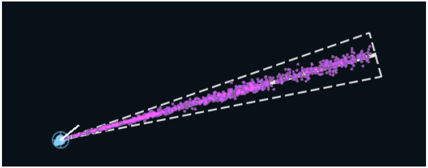
- **Introduction**
- **Sampling methods 101**
  - Rejection sampling
  - Metropolis-Hastings sampling
  - Inverse transform sampling
- **Extensions**
- **(Bonus) Mathematical details**

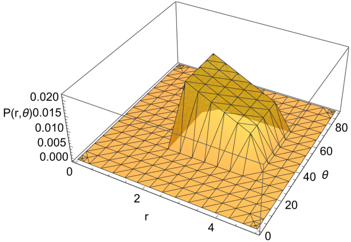
Introduction

Sampling: generating pseudo-random numbers that follow a given probability distribution.









Introduction

Given a probability distribution function (PDF) $f(x)$, we want to generate a bunch of points $\{x_i\}$ whose statistics are compatible with the PDF.

Crucial for any situation where we want a *discrete* representation of a *continuous* quantity.

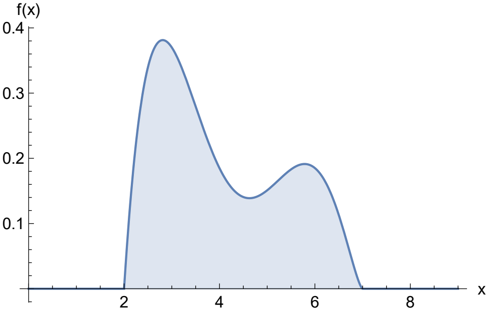
$$\textbf{True value:} \quad \mathbb{E}[\phi(X)] = \int \phi(x) f(x)\, dx$$

$$\textbf{Approximation:} \quad \tilde{\phi} = \sum_i \phi(x_i)$$

Notation: $X \sim f$ means that X is a random variable that is sampled from f.

Introduction

Given a PDF $f(x)$, how do we sample from it?



Easiest way: discretize and sample from discrete distribution. BUT, wildly inefficient - requires going through the entire PDF in fine detail!

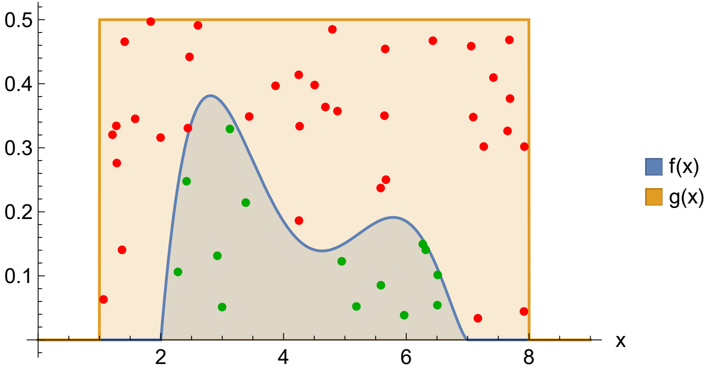
Goal today: sample **without** having to evaluate the function everywhere.

Sampling methods 101

Rejection Sampling

Main idea

Instead of sampling from $f(x)$, sample from a distribution that is easier. Reject any samples that are not in $f(x)$.



Rejection sampling

Let $f(x)$ be the distribution we want to sample from, and let $g(x)$ be a distribution we know how to sample from.

Rejection sampling algorithm:

1. Choose a constant C such that $Cg(x) \geq f(x)$ on all x.
2. Generate a sample $X \sim g$.
3. Generate a random number $U \sim \text{Unif}[0, 1]$.
4. Use U to determine if X is accepted or rejected.

$$U \leq \frac{f(X)}{Cg(X)} \implies \text{accept sample },$$
$$U > \frac{f(X)}{Cg(X)} \implies \text{reject sample }.$$

5. Repeat steps 2-4 until you have the samples you need.

Acceptance rate is $1/C$, the ratio of the areas under the curves of $f(x)$ and $Cg(x)$.

Rejection sampling

Pros:

- Only need to evaluate $f(x)$ on the generated points.
- Conceptually simple and computationally inexpensive.
- Works even if f, g aren't properly normalized.
- (Almost) any $g(x)$ will work***, "catch-all" method.

Cons:

- Need to have some idea of global behavior to pick C.
- Convergence can be slow for high-dimensional distributions.

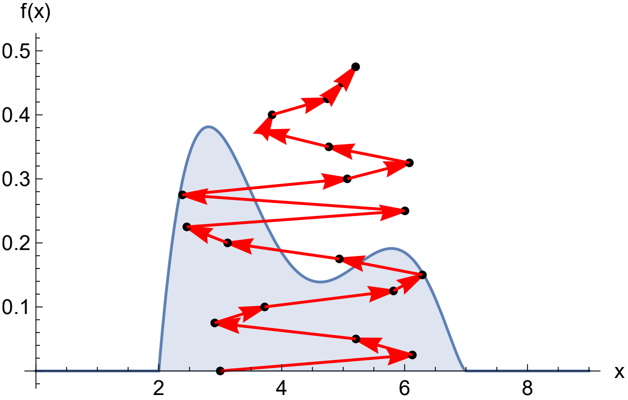
Typical choice for g is to use uniform distributions when f has finite support and Gaussians when f has infinite support.

***Some fall-off conditions required for infinite support

Metropolis-Hastings sampling

Main idea

Start from some point in $f(x)$, and do a random walk over many time-steps to generate samples from $f(x)$.



Metropolis-Hastings sampling

Let $f(x)$ be the distribution we want to sample from, and let $g(x'|x)$ be the "transition" distribution for the random walk.

Metropolis-Hastings sampling algorithm:

1. Pick an initial value x_0. Only requirement is $f(x_0) > 0$.
2. Generate a sample x' from $g(x'|x_0)$.
3. Compute the following:

$$\alpha = \min\left(1\,,\ \frac{f(x')}{f(x_0)}\frac{g(x_0|x')}{g(x'|x_0)}\right)$$

4. Generate a random number $U \sim \mathrm{Unif}[0, 1]$.
5. Determine whether we move to x' or stay at x_0:

$$U \leq \alpha \implies \text{use } x = x' \text{ as next sample}$$
$$U > \alpha \implies \text{use } x = x_0 \text{ as next sample}$$

6. Repeat steps 2-5, making sure to update the starting point to the newly-picked point x.

Metropolis-Hastings sampling

Intuition: $\alpha = 1$ means x' is more likely than x_0, so we go to x'. $\alpha < 1$ means x' is less likely than x_0, so we go to x' only some of the time.

Pros:

- Only need to evaluate $f(x)$ on the generated points
- Computationally inexpensive
- Works even if f, g aren't properly normalized

Cons:

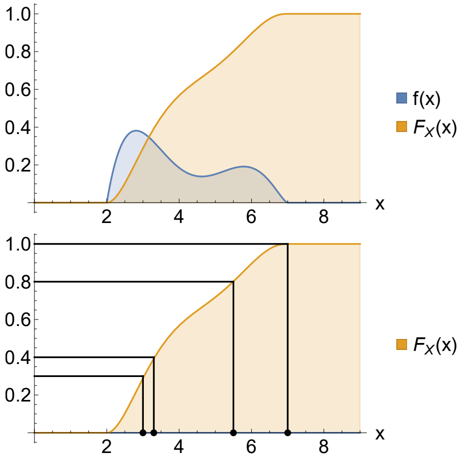
- Sensitive to initial conditions
- No easy way to tell when things have converged
- Difficult to converge when distribution is multi-modal

Typical choice for $g(x'|x)$ is to use a Gaussian centered at x.

Inverse transform sampling

Main idea

Compute the cumulative probability F_X by integrating f. Invert this to turn any $U \sim \text{Unif}[0, 1]$ into a sample $X \sim f$.



Inverse transform sampling

Let $f(x)$ be the distribution we want to sample from.

Inverse transform sampling algorithm:

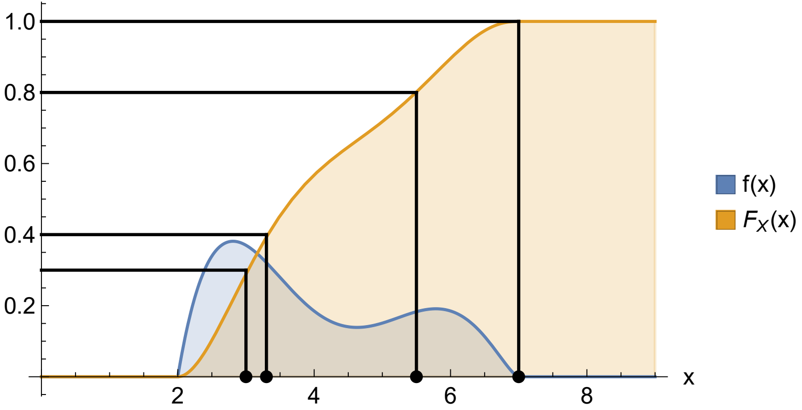
1. Compute the cumulative distribution function (CDF) $F_X(x) = \mathbb{P}(X \leq x)$ by integrating f:

$$F_X(x) = \int_{-\infty}^{x} f(x')\, dx'$$

2. Sample $U \sim \mathrm{Unif}[0, 1]$.
3. Compute the inverse CDF $F_X^{-1}(U)$ on U.
4. Set $X = F_X^{-1}(U)$ as your sample of $X \sim f$.
5. Repeat as many times as desired.

Inverse transform sampling

F_X increases monotonically and hits all values on $[0, 1]$, so it's a bijection. That means it's invertible!



Slope higher $\implies$ f is larger $\implies$ x should be more likely.

Inverse transform sampling

Pros:

- No need to worry about rejection or convergence.
- Can get away with only inverting F_X on the points U we generate.
- Closed-form solutions exist for linear and exponential distributions.

Cons:

- Need to integrate f over all values of x.
- Can almost never analytically invert F_X.
- Doesn't generalize well to more dimensions; would need to compute all partial integrals of f over all combinations of coordinates.

Extensions

Extensions

Rejection sampling:

- Adaptive rejection sampling - improve g dynamically by using information about where rejections are
- Ratio of transforms - if f is highly concentrated in one area, can "stretch" the function out to minimize rejections

Metropolis-Hastings sampling:

- Adaptive MCMC sampling - transition kernel is chosen dynamically to better explore the distribution. (Edges, multimodal distributions, etc.)
- Multiple-try Metropolis: pick multiple potential points to jump to at every step, compare likelihoods and pick a new point appropriately
- Other Markov chain Monte Carlo (MCMC) techniques

Inverse transform sampling:

- Piecewise distributions - pick which piece to sample from based on area, then apply inverse transform on that interval.
- Area-preserving maps

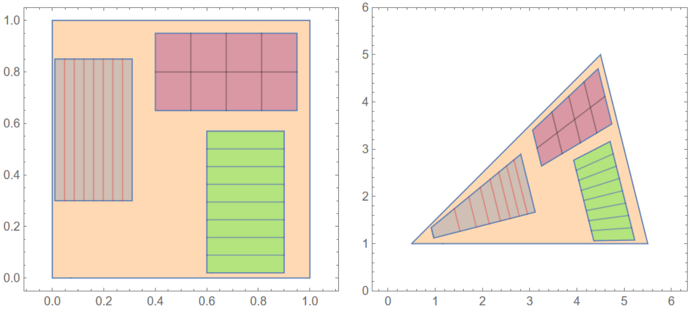
Area-preserving maps

Given a mapping from $\mathbb{R}^n$ to an n-polygon that preserves areas, you can uniformly sample the polygon in a closed-form way.

Example: suppose you have a triangle with vertices $\vec{v}_1$, $\vec{v}_2$, $\vec{v}_3$. Given two random variables $S, T \sim \text{Unif}[0,1]$, we can generate a point in the triangle as follows:

$$\vec{x}(S,T) = \vec{v}_1 + \sqrt{S}(\vec{v}_2 - \vec{v}_1) + T\sqrt{S}(\vec{v}_3 - \vec{v}_2)$$

Moreover, the map preserves areas, so $\vec{x}(S,T)$ is a uniform random point inside the triangle.



Mathematical details

Rejection sampling - mathematical details

We have $X \sim g$ and $U \sim \mathrm{Unif}[0,1]$, as well as a constant C such that $Cg(x) \geq f(x)$.

Probability of a sample being accepted:

$$\begin{aligned}
\mathbb{P}\left(X \text{ accepted}\right) &= \mathbb{P}\left(U \leq \frac{f(X)}{Cg(X)}\right) \\
&= \int \mathbb{P}\left(U \leq \frac{f(x)}{Cg(x)}\right) g(x)\,dx \qquad \text{[}X \sim g(x)\text{]} \\
&= \int \left(\frac{f(x)}{Cg(x)}\right) g(x)\,dx \qquad \text{[uniform prob. dist.]} \\
&= \frac{1}{C} \int f(x)\,dx \\
&= \frac{1}{C}\ .
\end{aligned}$$

Rejection sampling - mathematical details

Since $\mathbb{P}(X \text{ accepted}) = 1/C$, choosing $g^*(x) = Cg(x)$ to be as close to $f(x)$ as possible (i.e. smaller value of C) results in as large of an acceptance rate as possible.

Upper bound:

$$\mathbb{P}(X \text{ accepted}) = \frac{1}{C} \leq \frac{1}{\sup_x \left(\frac{f(x)}{g(x)}\right)} \, .$$

Bound on acceptance rate is determined *entirely* by the choice of the sample distribution g relative to the target distribution f.

Rejection sampling - mathematical details

Denote $F_X(t) = \mathbb{P}(X \leq t)$ be the CDF of X. Since X is a random variable with PDF f, we can obtain the CDF via simple integration:

$$F_X(t) = \int_{-\infty}^{t} f(x)\, dx \ .$$

We now want to find the CDF for all accepted samples we obtained from our rejection sampling algorithm:

Rejection sampling - mathematical details

$$
\begin{aligned}
&\mathbb{P}\left(X \le t \,|\, X \text{ accepted}\right) \\
&\quad = \frac{\mathbb{P}\left(X \le t\,,\, X \text{ accepted}\right)}{\mathbb{P}\left(X \text{ accepted}\right)} && \text{[Bayes' theorem]} \\
&\quad = C\,\mathbb{P}\left(X \le t\,,\, U \le \frac{f(X)}{Cg(X)}\right) && \text{[Use previous acceptance results]} \\
&\quad = C\,\mathbb{E}\left(\mathbf{1}\left\{X \le t\right\}\,\mathbf{1}\left\{U \le \frac{f(X)}{Cg(X)}\right\}\right) && \text{[Convert probability to expectation]} \\
&\quad = C\,\mathbb{E}\left(\mathbb{E}\left(\mathbf{1}\left\{x \le t\right\}\,\mathbf{1}\left\{U \le \frac{f(x)}{Cg(x)}\right\}\,\middle|\, X = x\right)\right) && \text{[Tower property]} \\
&\quad = C\,\mathbb{E}\left(\mathbf{1}\left\{X \le t\right\}\,\mathbb{E}\left(\mathbf{1}\left\{U \le \frac{f(x)}{Cg(x)}\right\}\,\middle|\, X = x\right)\right) && \text{[Pull X-only term out of $\mathbb{E}_U$]} \\
&\quad = C\,\mathbb{E}\left(\mathbf{1}\left\{X \le t\right\}\,\mathbb{P}\left(U \le \frac{f(x)}{Cg(x)}\,\middle|\, X = x\right)\right) && \text{[Convert expectation to probability]} \\
&\quad = C\,\mathbb{E}\left(\mathbf{1}\left\{X \le t\right\}\,\frac{f(X)}{Cg(X)}\right) && \text{[uniform pdf]} \\
&\quad = C\int\left(\mathbf{1}\left\{x \le t\right\}\,\frac{f(x)}{Cg(x)}\right) g(x)\,dx && \text{[Represent expectation as integral, $X \sim g$]} \\
&\quad = \int \mathbf{1}\left\{x \le t\right\}\, f(x)\,dx && \text{[cancel common terms]} \\
&\quad = \int_{-\infty}^{t} f(x) dx && \text{[integrand is non-zero only below t]} \\
&\quad = F_X(t)\ .
\end{aligned}
$$

Metropolis-Hastings sampling - mathematical details

[To be added later; sorry, I ran out of time. Ask me in-person to do the math on a blackboard and I'd be happy to oblige.]

Inverse transform sampling - mathematical details

Need to prove that setting $X = F_X^{-1}(U)$ for $U \sim \text{Unif}[0,1]$ yields a random variable whose CDF is in fact F_X. First, we write out the CDF of X as follows:

$$\mathbb{P}\left(X \leq x\right) = \mathbb{P}\left(F_X^{-1}(U) \leq x\right)$$

From here, we can use the fact that F_X is monotonic and continuous to say that we can apply F_X to both sides of the inequality $F_X^{-1}(U) \leq x$ and the inequality will still be true. Thus, we find that

$$\begin{aligned}\mathbb{P}\left(X \leq x\right) &= \mathbb{P}\left(F_X(F_X^{-1}(U)) \leq F_X(x)\right) \\ &= \mathbb{P}\left(U \leq F_X(x)\right) \\ &= F_X(x)\ ,\end{aligned}$$

where in the last line we used the fact that $\mathbb{P}(U < u) = u$ when U is sampled from the uniform distribution. So, we've proven that X has the desired CDF when using inverse transform sampling.