11/7-12: Simulation and Monte Carlo Methods

Overview

Last time we discussed the need for synthetic data.

Using synthetic data will typically require some form of simulation or random variate generation.

We'll discuss some of the basic methods.

Simulation

- Generating samples from a distribution to estimate some quantity
- Often don't know the distribution explicitly (normalizing constant)
- Used for generating data in synthetic data experiments
- Bayesian analysis; sampling from the posterior

Simulating Using the CDF

Suppose we have the cdf:

$$F(x) = \Pr(X \leq x)$$

In principle, can simply invert:

$$F^{-1}(p): \quad \Pr(X \leq F^{-1}(p)) = p$$

Then sampling amounts to generating $U \sim \text{Uniform}(0, 1)$ and outputing $F^{-1}(U)$.

Example

Suppose Y is an exponential r.v.

$$Y \sim \text{exp}(\lambda)$$

$$F(y) = \begin{cases} 1 - e^{-\lambda y} & y \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Then

$$F^{-1}(p) = \frac{\log(1-p)^{-1}}{\lambda}$$

Gaussian

Sampling from a Gaussian seems to be hard: inverse cdf $\Phi(x)$ can only be inverted numerically.

Clever solution is Box-Müller method; reminiscent of the trick to evaluate normalizing constant using polar coordinates

$$\int_{-\infty}^{\infty} e^{-\frac{1}{2}x^2}\, dx = \frac{1}{2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{1}{2}(x^2+y^2)}\, dx\, dy$$

Gaussian

Rather than a single sample, we generate *two*. Take $X_1 \sim$ Uniform$(0, 1)$ and $X_2 \sim$ Uniform$(0, 1)$.

Transform by

$$\begin{aligned} Y_1 &= \sqrt{\log(X_1^{-2})} \sin(2\pi X_2) \\ Y_1 &= \sqrt{\log(X_1^{-2})} \cos(2\pi X_2) \end{aligned}$$

Then (Y_1, Y_2) is a sample from a $\mathcal{N}(0, I)$ distribution.

Example: Beta

To sample from Beta(α_1, α_2):

$$\begin{aligned} X &\sim \text{Gamma}(\alpha_1, \beta) \\ Y &\sim \text{Gamma}(\alpha_2, \beta) \end{aligned}$$

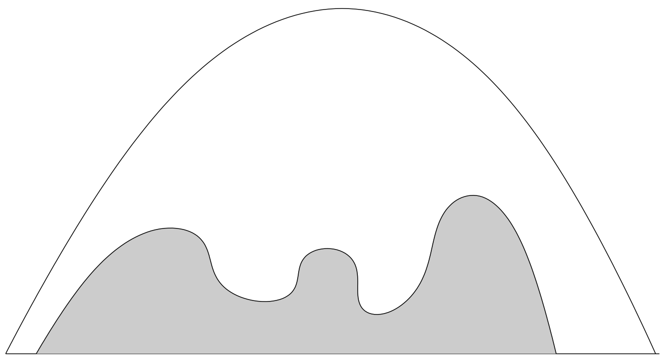
Then $\frac{X}{X+Y} \sim$ Beta(α_1, α_2).

Unfortunately, simulating a Gamma is "hard" — can't be done in 1 or 2 lines of code.

Example: Dirichlet

The same scheme generalizes to the Dirichlet.

Accept-Reject



Accept-Reject

Suppose we want to sample $X \sim f$ (density), and that we can come up with g, easy to sample from with $f(x) \leq Mg(x)$.

Intuitively, we sample the shaded area under the curve.

Accept-Reject

1. Generate $X \sim g$ and $U \sim$ Uniform$(0, 1)$.
2. If $U \leq \frac{f(X)}{Mg(X)}$, output X.
3. Otherwise, iterate

Accept-Reject: Analysis

Let Y be the r.v. that is output. We want to show we get the right c.d.f.

$$\begin{aligned}\Pr(Y \le y) &= \Pr\left(X \le y \,|\, U \le \frac{f(X)}{Mg(X)}\right) \\ &= \frac{\Pr\left(X \le y;\ U \le \frac{f(X)}{Mg(X)}\right)}{\Pr\left(U \le \frac{f(X)}{Mg(X)}\right)} \\ &= \frac{\int_{-\infty}^{y}\int_0^{f(x)/Mg(x)} du\, g(x)\, dx}{\int_{-\infty}^{\infty}\int_0^{f(x)/Mg(x)} du\, g(x)\, dx} \\ &= \frac{\frac{1}{M}\int_{-\infty}^{y} f(x)\, dx}{\frac{1}{M}}\end{aligned}$$

Accept-Reject: Analysis

Shows that probability of accepting is $\frac{1}{M}$.

Expected number of trials to get a sample: M. This may be too slow.

Also, need for $f \le Mg$ might be difficult to satisfy.

Discrete R.V. Simulation

In principle, inverting the c.d.f. does the trick. General method: if

$$F(x) = \begin{cases} 0 & x < t_1 \\ q_i & t_i \le x < t_{i+1} \\ 1 & x \ge t_n \end{cases}$$

then the following is the quantile function:

$$F^{-1}(p) = \begin{cases} t_1 & 0 < p \le q_1 \\ t_{i+1} & q_i < p \le q_{i+1} \\ t_n & q_{n-1} < p < 1 \end{cases}$$

Discrete R.V. Simulation

As before if $U \sim \text{Uniform}(0,1)$ then $F^{-1}(U) \sim F_X$.

[Picture of discrete cdf and inverse]

Example: Geometric

To simulate a geometric distribution, note

$$\begin{aligned}\Pr(X = k) &= (1-p)^{k-1}p \\ F(k) &= 1-(1-p)^k\end{aligned}$$

Slow way to simulate: Draw $U \sim$ Uniform.

$$\begin{aligned}\text{Is } U \le p\,? &\quad \Rightarrow 1 \\ \text{Is } U \le 1-(1-p)\,? &\quad \Rightarrow 2 \\ \text{Is } U \le 1-(1-p)^2\,? &\quad \Rightarrow 3\end{aligned}$$

Obviously inefficient...

Example: Geometric

Can invert by setting

$$K = \left\lfloor \frac{\log(1-U)}{\log(1-p)} \right\rfloor$$

to get a constant-time sampling algorithm.

General Sampling for Discrete Case

Could always compute the inverse cdf and get a $O(\log n)$ algorithm for sampling.

Can we get a constant time algorithm in general?

Surprising answer is "Yes" with a fairly elegant algorithm due to Walker (1974).

Alias Method

Suppose the distribution takes on n values.

Key idea: Write prob. mass function $f(x)$ as

$$f(x) = \frac{1}{n}(g_1(x) + g_2(x) + \cdots + g_n(x))$$

where each g_i is non-zero on only 1 or two values.

Alias Method

1. Sample $I \sim \text{Uniform}(1, 2, \ldots, n)$
2. Sample $X \sim g_I$

Requires $O(3)$ operations.

As we'll see, the setup cost is $O(n \log n)$.

Alias Method

How can the g_i be chosen?

There must be (x_1, y_1) with

$$\begin{aligned} f(x_1) &\leq \frac{1}{n} \\ f(y_1) &\geq \frac{1}{n} \end{aligned}$$

Define

$$g_1(x) = \begin{cases} nf(x_1) & x = x_1 \\ 1 - nf(x_1) & x = y_1 \\ 0 & \text{otherwise} \end{cases}$$

Alias Method

Now let f_1^* be what's left over: $f_1^*(x) = f(x) - \frac{1}{n} g_1(x)$, so that

$$f_1^*(x) = \begin{cases} 0 & x = x_1 \\ f(y_1) - g_1(y_1)/n & x = y_1 \\ f(x) & \text{otherwise} \end{cases}$$

Alias Method

Then f_1^* is supported on $n-1$ values, and sums to $\frac{n-1}{n}$. Thus, there must be (x_2, y_2) with

$$\begin{aligned} f_1^*(x_2) &\leq \frac{1}{n} \\ f_1^*(y_2) &\geq \frac{1}{n} \end{aligned}$$

Repeat the same argument!

In this way, we "peel away" two elements at a time. By sorting, we can implement this in $O(n \log n)$ time.

Monte Carlo Sampling

Suppose that we want to estimate an expectation

$$E[h] \quad = \quad \int_{\mathcal{X}} h(x)\, f(x) dx$$

Natural Monte Carlo method is to generate $X_1, X_2, \ldots, X_N$ with $X_i \sim f$ and then

$$E[h] \quad \approx \quad \frac{1}{N} \sum_{i=1}^{N} h(X_i)$$

This gives an unbiased estimate of $E[h]$.

Monte Carlo Sampling

What if f is difficult to sample from?

Importance Sampling

Idea is to sample instead from an "easier" density g that is (hopefully) close to f.

Then, the ith sample by the importance weight

$$w_i \quad = \quad \frac{f(X_i)}{g(X_i)}$$

Importance Sampling

The fact that this works follows from

$$\begin{aligned} E[h] \quad &= \quad \int h(x)\, f(x)\, dx \\ &= \quad \int h(x)\, \frac{f(x)}{g(x)}\, g(x)\, dx \end{aligned}$$

(As long as g is not zero where f is non-zero.)

Importance Sampling

Monte Carlo approximation to the integral is

$$\begin{aligned} E[h] &\approx \frac{1}{N}\sum_{i=1}^{N} h(X_i)\,\frac{f(X_i)}{g(X_i)} \\ &= \frac{1}{N}\sum_{i=1}^{N} h(X_i)\,w_i \end{aligned}$$

where $X_i \sim g$. Coming up with good densities g can be difficult, and is sometimes still the subject of research.