rejectionsampling

May 8, 2023

1 Rejection Sampling

First, consider the case for a continuous probability distribution: Let X be a random variable with density f. The function F is continuous and only takes positive values on $[a, b]$ except for 0. Since continuity holds, we can determine the maximum. The goal of this method is to generate a random variable with such density. A random point is generated from the uniform distribution on a rectangle. If it is below f, it is accepted; otherwise, it is rejected. Executing the algorithm involves generating a random variable X with density f. A random variable Y is efficiently generated with density g. Importantly, Y should take values from the same set as X. A constant c is determined such that:

$$\sup_{x\in\mathbb{R}} \frac{f(x)}{g(x)} = c \leq \infty.$$

Ideally:

$$\sup_{x\in\mathbb{R}} \frac{f(x)}{g(x)} \leq c \leq \infty.$$

If $U \leq \frac{f(Y)}{c\cdot g(Y)}$,then X is returned as Y. If not, another Y will be generated again. In the example below, the distribution Semicircle(1) with Uniform(-1,1) is presented, and two probability density functions (PDFs), semicircle_pdf and uniform_pdf, are defined. When the range of x extends beyond $[-1, 1]$, the PDF value is set to 0. The generator is tested by calling the test_generator function. The case for a discretize probability distribution: Let CDF, F for X:

$$F(x) = \sum_{i\leq j} p(x_i)$$

Discrete random variables can be generated by slicing up the interval $(0, 1)$ into subintervals which define a partition of $(0, 1)$:

$$(0, F(x_1)), (F(x_2), F(x_3)), \dots, (F(x_{k-1}), 1)$$

generating random variables U from a uniform distribution on the interval (0, 1) and determining the subinterval into which U falls. A realization of a discrete random variable X is generated according to the given formula:

$$(X = i) = p_i,\ i = 1, 2, \dots, n.$$

It is possible to generate another random variable Y with a distribution defined by the formula:

$$\mathbb{Y}(Y = i) = q_i,\ i = 1, 2, \dots, n.$$

Then, a random variable U is generated from the uniform distribution $\mathcal{U}(0,1)$, such that U is independent of Y. If U is evaluated to satisfy the condition $U \leq \frac{pY}{c \cdot q_y}$, where $c = \max_i \frac{p_i}{q_i}$, then X is set to be equal to Y. Otherwise, we go back to the beginning of the algorithm and generate Y again.

In the example below, rejection sampling is used for a discrete probability distribution. The Bernoulli distribution $\mathcal{Be}(\frac{1}{4})$ with Uniform(0,1) is chosen. In the code, two probability mass functions (PMFs) are defined: bernoulli_pmf and uniform_pmf. The bernoulli_pmf(x) function returns the PMF value based on the input x: $\frac{3}{4}$, $\frac{1}{4}$, and zero for $x = 0$, $x = 1$, and any other value respectively. Similarly, the uniform_pmf(x) function returns the PMF value based on the input x: $\frac{1}{2}$ for $x = 0$ or $x = 1$, and zero for any other value. Then, the generator is tested by calling the test_generator function.