Lecture 3

Sampling 1D Distributions

Contents

1 Goal

In this lecture, we shall learn two techniques to to sample from **arbitrary 1D distributions**. Both of these techniques rely on the continuous uniform distribution $U[0,1]$ as their engine.[1]

(i) Transformation Method

(ii) Accept-Reject Method[2]

2 Transformation Method

The idea in this method is to use a sequence of random numbers from $U[0,1]$

$$U_1, U_2, ..., U_n \sim g(u) = U[0,1],$$

to produce a sequence of random numbers from the target PDF,

$$X_1, X_2, ..., X_n \sim f(x),$$

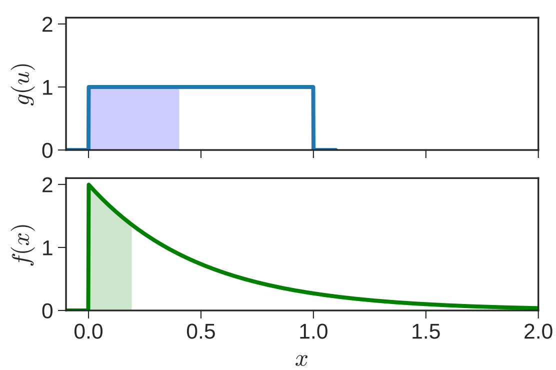
by seeking a mathematical **mapping** or **transformation** $(u \to x)$. This mapping may be expressed as a function $x(u)$.

Before I introduce the method, note that:

- the relationship between CDF and PDF is unique
- the mapping $x(u)$ will depend on $f(x)$

2.1 Derivation

First, let me try to illustrate the idea graphically. Consider the PDFs shown below. At the top is the uniform distribution $U[0,1]$.



At the bottom is some arbitrary distribution $f(x)$ that we seek to sample from. Note that the domains of $g(u)$ and $f(x)$ need not be the same.

[1] Previously we learned how to sample from $\mathbf{U}[0,1]$ using the LCG method. Superior methods were alluded to, but not discussed in detail.

[2] We already met this method, when we tried to find the area of a circle by throwing darts. There we used it to integrate functions; here we shall generalize it to address the sampling problem.

Intuition: Given U, we pick X so that the shaded blue and green areas are the same

We need to put this intuition into a mathematical framework.

$$\begin{aligned} \text{Pr}(U \leq u) &= \text{Pr}\left(X \leq {\color{red}x(u)}\right). \\ \int_{-\infty}^{u} g(U)\, dU &= \int_{-\infty}^{x(u)} f(X)\, dX \\ u &= F(x(u)), \end{aligned} \tag{1}$$

where $F(x)$ is the CDF corresponding to $f(x)$.

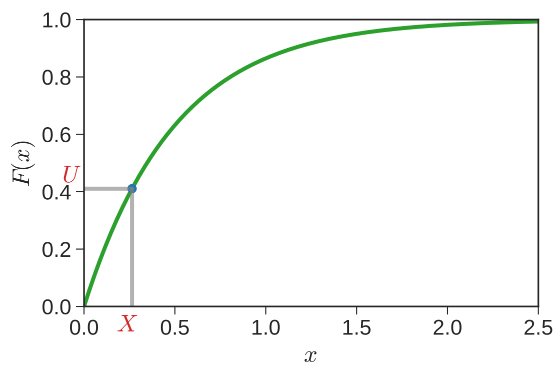
Thus, the transformation method boils down to this: given the target PDF $f(x)$, find the corresponding CDF $F(x)$, by trying to solve,

$$\boxed{u = F(x)} \tag{2}$$

in terms of x. If it is possible to solve explicitly for $x = F^{-1}(u)$ in terms of u, then we are in business!

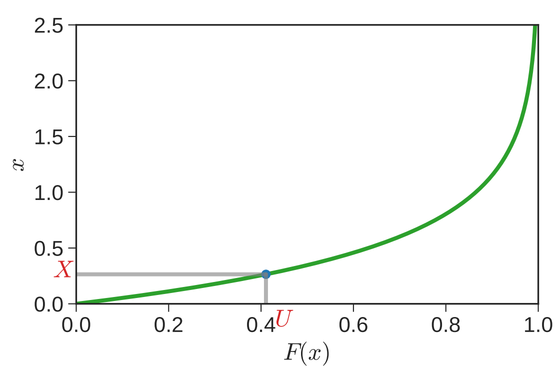
2.2 Intuition

The **domain** of $U[0,1]$ matches the **range** of the $F(x)$, viz. $[0,1]$.



- draw $U \sim U[0,1]$ [pick a point on the y-axis in the figure above],
- use the CDF (solve $F(X) = U$) to get X

It takes a while to internalize this insight. It may be helpful for some of you to flip the picture above on its side.



Exercise: Based on the plot, can you qualitatively see why we are more likely to sample for $x < 1.0$ than $x > 1.0$?

The success of this method hinges on our ability to efficiently solve for $x = F^{-1}(u)$. If we can analytically solve $x = F^{-1}(u)$, this is likely our best option. Alternatively, we may be able to use some sort of piecewise curve (linear, cubic, spline) to fit $F(x)$, and then use inverse interpolation. But, this will incur a cost. In general, we could use a nonlinear solver, but that could involve even more overhead.

Enough theory. Let us do an example.

Example: Generate random numbers, x_i, from the exponential distribution with $\lambda > 0$:

$$f(x;\lambda) = \lambda \exp(-\lambda x), \quad x > 0.$$

Solution: According to the transformation method:

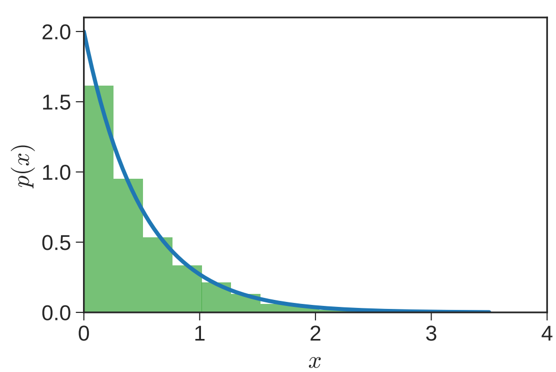
$$\begin{aligned} u &= F(x) \\ &= \int_0^x \lambda \exp(-\lambda X)\, dX \\ u &= 1 - e^{-\lambda x}. \end{aligned}$$

Thus, we can solve for x in terms of u,

$$x = \frac{-\ln(1-u)}{\lambda}.$$

The program `myExponentialSampler(Lambda, NumSamples)` in sec. A.2.1 implements this transformation. Suppose $\lambda = 2$. Let us generate 5000 numbers and histogram them.

```
x = myExponentialSampler(2.0, 5000)
```



The blue line is the actual distribution $2\exp(-2x)$.

Example: Consider a PDF nonzero over $x = [0, 1]$, given by,

$$f(x) = \begin{cases} 8x & 0 \le x < 0.25 \\ \dfrac{8}{3}(1-x) & 0.25 \le x \le 1 \end{cases} \tag{3}$$

Find the transformation $x(u)$ required to sample from $f(x)$.

Solution: The CDF of $f(x)$ is found to be,

$$F(x) = \begin{cases} 4x^2 & 0 \leq x < 0.25, \\ \dfrac{8}{3}\left(x - \dfrac{x^2}{2} - \dfrac{1}{8}\right) & 0.25 \leq x \leq 1, \end{cases} \tag{4}$$

When $x = 0.25$, $u = 4(0.25)^2 = 0.25$. Therefore, $x = F^{-1}(u)$ is given by,

$$F^{-1}(u) = \begin{cases} \dfrac{\sqrt{u}}{2} & 0 \leq u < 0.25, \\ 1 - \dfrac{\sqrt{3(1-u)}}{2} & 0.25 \leq u \leq 1. \end{cases} \tag{5}$$

2.3 Discrete Distributions

Suppose, X is a random variable that takes discrete values $\{x_0, x_1, x_2, \cdots, \} = \{0, 1, 2, \cdots\}$[3]. Let the CDF be given by $F(x_i) = F_i$. In this case, the transformation method may be suitably adapted.

We generate a uniform random number $U \in U[0, 1]$, and select the smallest x_i for which $F_i \geq U$.

$$X = \min\{x_i \mid U \leq F_i\} \tag{6}$$

In general, this requires a search to find the minimum.

Example: Consider a coin toss,

$$f(x) = \begin{cases} 0.5, & x = x_0 = 0 \\ 0.5, & x = x_1 = 1. \end{cases}$$

The CDF is $F(x_0 = 1) = 0.5$ and $F(x_1 = 1) = 1.0$ Thus, the transformation method yields,

$$X = \begin{cases} x_0 = 0, & U \leq F(x_0) = 0.5 \\ x_1 = 1, & F(x_0) < U \leq F(x_1), \end{cases}$$

which can be simplified as,

$$X = \begin{cases} 0, & U \leq 0.5 \\ 1, & 0.5 < U \leq 1. \end{cases}$$

2.4 Summary

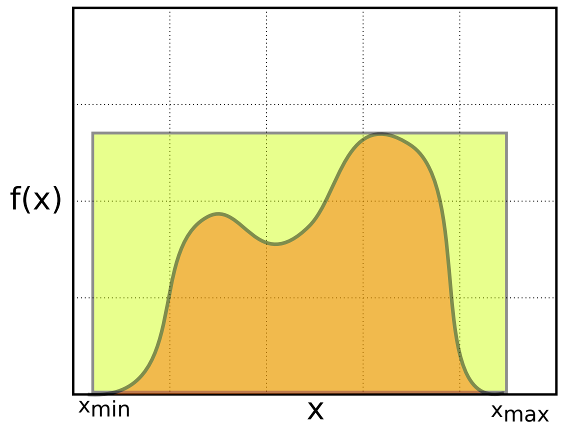
- (-) requires a closed form expression for $F(x)$; or potentially costly numerics
- (-) may not always be the simplest or fastest for a particular target distribution (example Gaussian distribution)
- (-) can become unworkable for non-standard multidimensional PDFs
- (+) generally works quite fast
- (+) works for any monotonically increasing $F(x)$ (could be flat in some regions)

[3]the domain may be finite, or infinite; we assume x_i are sorted in ascending order.

- (+) can be used for continuous or discrete distributions

3 Accept-Reject Method

The idea is to enclose the distribution in an appropriate box, and throw darts at it.



We used this method to compute integrals. From a MC perspective, integration $\subset$ sampling.[4]

For simplicity, suppose the distribution $f(x)$ has a finite domain $x_{\min} \leq x \leq x_{\max}$. We can draw a box of height $f_{\max}$ which encloses the distribution.

3.1 Algorithm

Then algorithm then works as follows:

- pick $X \sim U[x_{\min}, x_{\max}]$
- pick another random number $U \sim U[0, f_{\max}]$
- if $U \leq f(X)$, then accept X; if not, reject X and repeat.

The set of accepted samples follows the target distribution.

Example: Consider the distribution

$$f(x) = \frac{3}{8}(1 + x^2), \quad -1 \leq x \leq 1$$

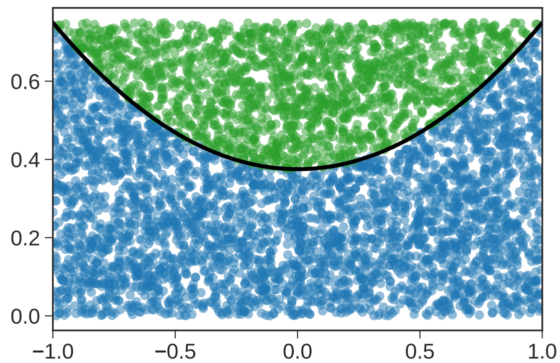
If we plot the function, we realize that the maximum value occurs at $x = \pm 1$, thus $f_{\max} = f(\pm 1) = 3/4$. This method is easy to implement.

```
def acceptRejectExample(ndarts):

    xmin = -1.; xmax = 1.; fmax = 3./4

    x = np.random.uniform(xmin, xmax, ndarts)
    u = np.random.uniform(0., fmax, ndarts)

    y = x[u <= 3./8. * (1 + x**2)]
```

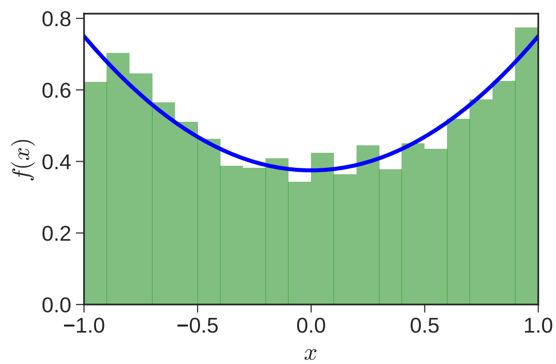
[4] recall big picture!

```
    return y

x = acceptRejectExample(5000)
```

The figure below shows the location of all the darts. A subset (green) is rejected, while the rest (blue) are returned.



We can consider the histogram of the sampled points x, where the blue line is the target distribution.

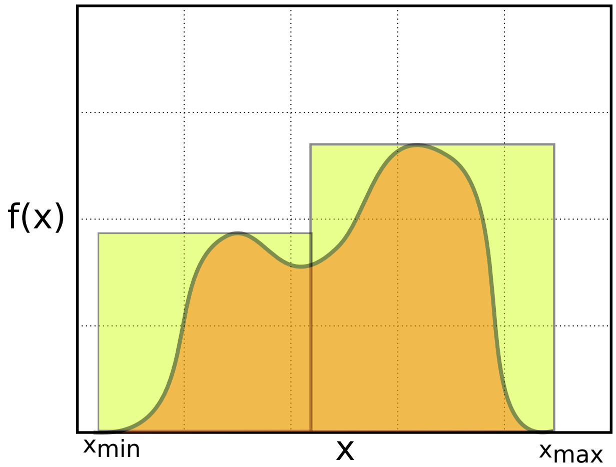


Exercise: Solve this problem using the transformation method.

3.2 Summary

Accept-Reject is a general method that can be applied to more than 1D.

- the target distribution need not be invertible
- even if the domain is infinite, the computational domain (e.g. floats) is not infinite
- if the distribution is sharply peaked we may draw multiple boxes of different heights to reduce rejection



However, we have to be careful not to oversample from the smaller box. Set the probability of picking X from a particular box proportional to the area of the box (see section A.1.)

Adaptive Rejection Sampling is a modification to the rejection sampling method that tries to improve the basic algorithm. We will not consider it in detail in this class, but it essentially:

- defines the envelope distribution in log space (e.g. log-probability or log-density)
- uses a piecewise linear density function as the envelope
- The envelope is adapted as sampling is carried on

ARSpy is a pure python package that can perform adaptive rejection sampling in 1D.

4 Epilog

For 1D sampling, one of these two methods should generally suffice.

As the number of dimensions increases:

- the **transformation method** gets increasingly hard to apply, even if we use numerical rather than analytical inversion
- the number of rejections in the **accept-reject method** increases rapidly

Even sophisticated variants of these simple algorithms can no longer cope with this complexity. Next, we shall look at 2D distributions, as a window into these higher dimensional spaces.

As the dimensionality and complexity increase, MCMC methods - which are not particularly competitive for "simple textbook problems" - start to shine!

5 Problems

5.1 Thought Questions

(i) Use the transformation method to sample from the following distributions:

(a) Continuous $U[-1, 1]$,

$$f(x) = \begin{cases} 0.5 & -1 \leq x \leq 1 \\ 0 & \text{elsewhere} \end{cases}$$

(b) Discrete Coin Toss,

$$f(x) = \begin{cases} 0.5, & x = 0 \\ 0.5, & x = 1 \end{cases}$$

(ii) Use the transformation method to sample from the distribution:

$$f(x) = \begin{cases} 2(1 - x) & 0 \leq x \leq 1 \\ 0 & \text{elsewhere} \end{cases}$$

(iii) Suppose you are given a function which already samples from the exponential distribution

$$g(y) = \lambda_1 \exp(-\lambda_1 y).$$

Can you use the transformation method and this distribution to sample from another exponential distribution:

$$f(x) = \lambda_2 \exp(-\lambda_2 x).$$

Hint: Match the CDFs.

(iv) Solve any integration problem using a bad RNG like RANDU, and any sophisticated RNG that passes the usual tests. Scour the web to find any real or physical simulation that was adversely affected by poor choice of RNG. Report your findings.

(v) Consider sampling from the Beta and Gamma distributions using the transformation method. You will need the special functions: inverse incomplete Beta, and inverse incomplete Gamma functions, respectively.

5.2 Numerical Problems

(i) Discrete Distribution

Consider a discrete probability distribution

$$\begin{aligned} f(x = 1) &= 0.50 \\ f(x = 2) &= 0.25 \\ f(x = 3) &= 0.15 \\ f(x = 4) &= 0.10 \end{aligned}$$

Use the transformation method to sample from $f(x)$. Generate 10^5 samples, and plot a histogram.

(ii) Transformation Method

Consider a linear PDF

$$f(x) = 2x, \quad 0 \leq x \leq 1 \tag{7}$$

Use the transformation method to generate 10^5 samples from $f(x)$. Plot a histogram of the samples and compare it with the PDF.

(iii) Piecewise CDF

Consider the Laplacian distribution,

$$f(x) = \frac{1}{2}e^{-|x|}, \quad -\infty < x < \infty \tag{8}$$

(a) Find the CDF of $f(x)$. **Hint**: Split into $x < 0$ and $x > 0$ cases.

(b) Use the transformation method to develop a mapping $x(u)$ to sample $f(x)$. **Note**: You do not have to implement the method.

(iv) Rayleigh Distribution

Given a mapping using the transformation method,

$$x(u) = \sqrt{-2\log(1-u)} \tag{9}$$

where $u \sim U[0, 1]$.

(a) Show that the domain of x is $[0, \infty]$.

(b) Find the corresponding PDF.

(v) Inverse Interpolation

Consider the distribution

$$f(x) = 6x(1-x), \;\; 0 \leq x \leq 1 \tag{10}$$

Generate $\approx 10^5$ samples from this distribution using all different methods suggested below. Test your solution by plotting the histogram of the samples and $f(x)$.

(a) the accept-reject method with an appropriate f_{max}.

(b) the transformation method: you can:

   i. numerically solve any nonlinear equation that may arise, *or*

   ii. use "inverse linear interpolation"

   - Generate a F versus x table, $\{F_i = F(x_i), x_i\}$, where $F(x)$ is the CDF
   - Generate a uniform random number $u \sim U[0, 1]$
   - Use the table and linear interpolation to find the x corresponding to $F = u$

(vi) Expected Values

Suppose you are an insurance company that operates in north Florida, and modeling worst case scenarios is important for your business. After poring through historical data and climate models, you determine that the probability of a major hurricane in a given year is $p = 0.1$. We can represent the occurrence of a hurricane y by a discrete distribution,

$$y = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } 1-p \end{cases}$$

When a major hurricane does hit, the loss z (in billion \$) is exponentially distributed with $\pi(z) = \lambda e^{-\lambda z}$, with $\lambda = 0.3$. Thus, the loss for any year is given by the random variable $x = yz$,

$$\pi(x) = \begin{cases} \lambda e^{-\lambda x} & \text{with probability } p \\ 0 & \text{with probability } 1-p \end{cases}$$

(a) Find the expected loss $E(x) = \int x\pi(x)dx$ in any given year analytically.

(b) Your firm is conservative, and you keep reserves of $\mu = 10$ (in billion \$) to pay out claims in the event of a disaster. Suppose what you really care about is the expected loss x, only when it exceeds the threshold μ. Mathematically, you wish to compute the *expected shortfall*:

$$E(x|x > \mu) = \int_{x>\mu} xq(x)dx,$$

where $q(x) \sim \pi(x)$ for $x > \mu$, and $q(x) = 0$, elsewhere.

Use Monte Carlo to sample from $q(x)$ directly. Hence, determine the expected shortfall, and associated error.

(vii) Uncertainty Analysis by MC Simulation

Consider a formula for net present value, which arises from a discounted cash flow analysis,

$$p = \sum_{i=1}^{n} \left(\frac{1+g}{1+r}\right)^i,$$

where the discount rate $r = 0.10 \pm 0.03$, the growth rate $g = 0.05 \pm 0.02$, and $n = 10$.

(a) What is the expected value of p? Estimate the error in p, σ_p, assuming r and g are independent.

(b) Generate 10^6 samples of $r \sim \mathcal{N}(0.1, 0.03)$, and $g \sim \mathcal{N}(0.05, 0.02)$, to generate 10^6 realizations of p. What is the mean and standard deviation of p?

(c) Compare the two estimates, and comment.

(viii) Central Limit Theorem: Draw n samples X_i from the exponential distribution,

$$X_i \sim \pi(x) = \lambda e^{-\lambda x},$$

with $\lambda = 1/2$.

(a) Theoretically, evaluate the mean and variance of $\pi(x)$.

(b) Define the sample mean

$$Y = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

Generate 10^6 samples of Y for different values of $n = \{3, 5, 10, 50, 100\}$.

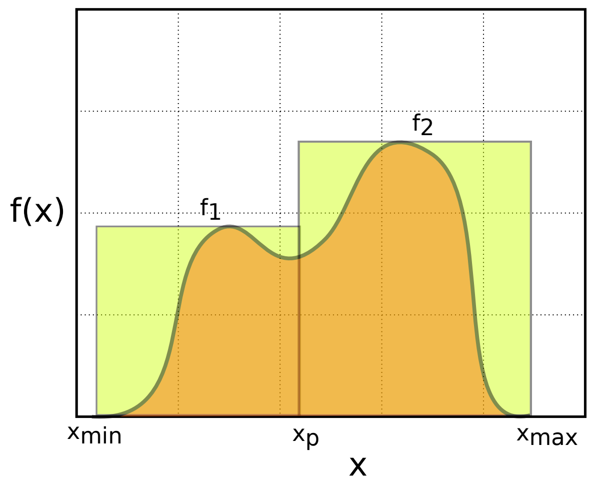
(c) Plot the histograms of Y for the different n.

(d) Explain your observations in terms of the CLT.

A Appendix

A.1 Rejection Sampling Distributions with Two Peaks

Suppose the heights of the two boxes are f_1 and f_2, and x_p partitions the domain into two parts.



- The corresponding areas $A_1 = (x_p - x_{\min})f_1$ and $A_2 = (x_{\max} - x_p)f_2$. First we decide which of the boxes to select.[5] The discrete probability distribution is given by:

$$\text{pick } X \sim \pi(x) = \begin{cases} \dfrac{A_1}{A_1 + A_2}, & \text{box 1} \\ \dfrac{A_2}{A_1 + A_2}, & \text{box 2} \end{cases}$$

  To do this, draw $U \sim U[0, 1]$. If it is less than $A_1/(A_1 + A_2)$, then select box 1, and vice versa.

- depending on the box, sample X uniformly between $[x_{\min}, x_p]$ or $[x_p, x_{\max}]$

- now pick U according to, pick

$$U \sim \begin{cases} U[0, f_1] & X < x_p \\ U[0, f_2] & X \geq x_p \end{cases}.$$

- If $U \leq f(X)$, then accept X. If not, reject X and repeat.

A.2 Python Programs

A.2.1 Exponential Sampler

```
def myExponentialSampler(lam, num):
    """Generate -num- exponentially distributed samples"""
    u = np.random.random((num,1))
    x = -np.log(1-u)/lam
    return x
```

[5] this is identical to simulating a biased coin