Algorithms for random variates generation

Giacomo Babudri

December 2023

1 Introduction

1.1 Random Variate

In probability and statistics, a random variate or simply variate is a particular outcome or realization of a random variable; the random variates which are other outcomes of the same random variable might have different values (random numbers).

A random deviate or simply deviate is the difference of a random variate with respect to the distribution central location (e.g. mean), often divided by the standard deviation of the distribution (i.e. as a standard score)

Random variates are used when simulating processes driven by random influences (stochastic processes). In modern applications, such simulations would derive random variates corresponding to any given probability distribution from computer procedures designed to create random variates corresponding to a uniform distribution, where these procedures would actually provide values chosen from a uniform distribution of pseudorandom numbers.

1.2 Random Variates generation

In the expansive landscape of statistical and computational methodologies, the generation of random variates stands as a critical facet influencing the accuracy and reliability of simulations and models. As computational models become increasingly sophisticated and diverse, the need for advanced techniques in random variate generation becomes paramount. This thesis embarks on a comprehensive exploration of advanced approaches to random variate generation, delving into methodologies that bridge theoretical foundations with practical applications.

An example of definition based on what Devroye defines a random variate generation algorithm (for real numbers):
Assume that

1. Computers can manipulate real numbers.

2. Computers have access to a source of random variates that are uniformly distributed on the closed interval [0,1].

Then a random variate generation algorithm is any program that halts almost surely and exits with a real number x. This x is called a random variate.

(Both assumptions are violated in most real computers. Computers necessarily lack the ability to manipulate real numbers, typically using floating point representations instead. Most computers lack a source of true randomness (like certain hardware random number generators), and instead use pseudorandom number sequences.)

2 General Methods

In the context of random variate generation, a method represents a systematic approach to obtaining random values that follow a specific probability distribution.

2.1 Inverse Transform Method

The Inverse Transform approach is one of the fundamental methods for generating random variates from a continuous probability distribution. This method exploits the inverse relationship between the cumulative distribution function (CDF) and the variable of interest.

Procedure:

1. **Probability Distribution:** Consider a continuous probability distribution with a probability density function (pdf) $f(x)$ and a cumulative distribution function (CDF) $F(x)$.

2. **Uniform Random Variable:** Generate a uniform random variable U distributed in the interval $(0, 1)$.

3. **Application of the Inverse:** Apply the inverse of the CDF to the uniform random variable U to obtain a random variable X distributed according to the desired probability distribution.

Mathematical Formula:

$$X = F^{-1}(U)$$

Where:

- X is the desired random variable.

- U is a uniform random variable in $(0, 1)$.
- F^{-1} represents the inverse function of the CDF.

Example:
Consider the exponential distribution with probability density function $f(x) = \lambda e^{-\lambda x}$ and CDF $F(x) = 1 - e^{-\lambda x}$.
The inverse of the CDF is $F^{-1}(U) = -\frac{\ln(1-U)}{\lambda}$, where U is a uniform random variable.

2.2 Convolution Method

The Convolution method is a technique that involves combining independent random variables to obtain a distribution for the sum of these variables. The method is particularly relevant when dealing with the distribution of the sum of independent and identically distributed random variables.

Procedure*:*

1. **Probability Distributions:** Consider n independent random variables $X_1, X_2, \ldots, X_n$ with probability density functions (pdfs) $f_1(x), f_2(x), \ldots, f_n(x)$ and cumulative distribution functions (CDFs) $F_1(x), F_2(x), \ldots, F_n(x)$, respectively.

2. **Convolution Operation:**

   Consider n independent random variables $X_1, X_2, \ldots, X_n$ with probability density functions (pdfs) $f_1(x), f_2(x), \ldots, f_n(x)$ and cumulative distribution functions (CDFs) $F_1(x), F_2(x), \ldots, F_n(x)$.

   The probability density function of their sum $Y = X_1 + X_2 + \ldots + X_n$ is given by the convolution:

$$f_Y(y) = \int_{-\infty}^{\infty} f_1(y - x_1) f_2(y - x_2) \ldots f_n(y - x_n)\, dx_1 dx_2 \ldots dx_n$$

3. **Algorithmic Procedure:** In practice, the convolution operation is often implemented using the convolution theorem in the frequency domain or numerical techniques such as discrete convolution.

Example: Convolution of Exponential distributions
Let X and Y be two independent random variables with exponential pdfs:

$$f_X(x) = \lambda e^{-\lambda x} \quad \text{and} \quad f_Y(y) = \lambda e^{-\lambda y}$$

The convolution of these distributions gives the pdf of the sum $Z = X + Y$:

$$f_Z(z) = \int_0^z \lambda e^{-\lambda(z-y)} \lambda e^{-\lambda y}\, dy$$

Solving this integral provides the pdf of the convolution distribution $f_Z(z)$.

2.3 Acceptance-Rejection Method:

The Acceptance-Rejection method is a technique used in random variate generation to simulate random samples from a desired probability distribution. The basic idea is to generate candidate samples from a simple, easily invertible distribution (the proposal distribution) and accept or reject them based on a comparison with the desired distribution. If a candidate sample falls under the desired distribution, it is accepted; otherwise, it is rejected, and the process is repeated.

Procedure:

1. **Proposal Distribution** Choose a simple and easily invertible probability distribution, known as the proposal distribution. This distribution should envelop the target distribution.

2. **Generate Candidate Samples**

   Generate random samples from the proposal distribution. These samples serve as candidates for the desired distribution.

3. **Evaluate Acceptance Criterion**

   For each candidate sample, calculate the acceptance criterion by comparing the ratio of the probability density function (pdf) of the desired distribution to the proposal distribution.

4. **Accept or Reject**

   Accept the candidate sample if a randomly generated value from a uniform distribution is less than the acceptance criterion; otherwise, reject the sample.

5. **Repeat as Needed**

   Repeat steps 2-4 until a sufficient number of accepted samples is obtained.

Algorithm 1 Acceptance-Rejection Method

function ACCEPTANCEREJECTION(target_pdf, proposal_pdf, num_samples)
  accepted_samples $\leftarrow$ []
  while length(accepted_samples) $<$ num_samples **do**
    Generate a candidate sample x from the proposal distribution
    Calculate the acceptance criterion: $u \sim \text{Uniform}(0, 1)$
    $A \leftarrow \frac{\text{target\_pdf}(x)}{\text{proposal\_pdf}(x)}$
    if $u < A$ **then**
      accepted_samples.append(x)
    end if
  end while
  return accepted_samples
end function

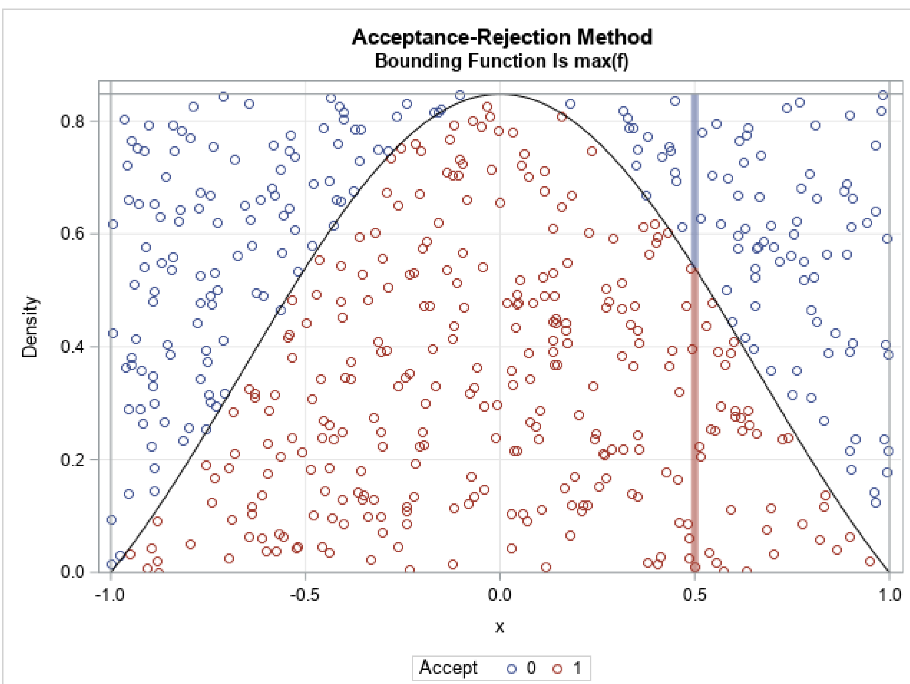

Figure 1: Simulation of Acceptance and Rejection Method