Simulation of Random Variables

MATH118 Spring 2022

Serdil Anıl Ünlü Ahmet Tuğkan Ayhan Sena Nur Ulukaya Yunus Emre Öztürk
Gökbey Gazi Keskin

Gebze Technical University

Contents

Random Number Generators

Introduction

[1]
Random numbers: Numbers that occur in a sequence such that two conditions are met:

1)The values are uniformly distributed over a defined interval or set.

2)It is impossible to predict future values based on post or present ones.

Random Number Generator

Question: What is the Random Number Generator?

Random Number Generator

Question: What is the Random Number Generator?

- The earliest methods for generating random numbers such as dice, coin flipping and roulette wheels, are still used today, mainly in games and gambling as they tend to be too slow for most applications in statistics and cryptography.

- Random number generators have applications in gambling ,statistical sampling, computer simulation,cryptography, completely randomized design,and other areas where producing an unpredictable result is desirable.

- Random number generators are very useful in developing **Monte-Carlo** method simulations.

Type of Random Number Generators:

-There are 2 principal methods used to generate random numbers

- **First Method (True Random Numbers)**
- **Second Method (Pseudo Random Numbers)**

What is the True Random Numbers?

First Method (True Random Numbers)

Definition: Some physical phenomenon is measured that is expected to be random and then compensates for possible biases in the measurement process. For example, cosmic background radiation or radioactive decay as measured over short timescales represent sources of natural entropy.

Second Method (Pseudo Random Numbers)

What is the Pseudo Random Numbers?

Definition: Computational algorithms are used that can produce long sequences of apparently random results, which are in fact completely determined by a shorter initial value, known as a seed value or key.

As a result , the entire seemingly random sequence can be reproduced if the seed value is known.

Second Method (Pseudo Random Numbers)

-**Pseudo-Random Number Generators**,which generate numbers that look random, but are actually deterministic, and can be reproduced if the state of the **Pseudo-Random Number Generators.**

- Several computational methods for pseudo-random number generation exist.

- All fall short of the goal of true randomness , although they may meet, with varying success. - Most of the algorithms for generating pseudo random numbers are recursive in nature. They may be expressed as: $[X_{i+1} = f(x_i)]$

- Where,$[X_i]$ is the number lying between 0 and 1.

-The innovation lies in choosing the form of the function f(x).

-For a uniform distribution on [0,1], the function should be such that it should lie on the square of unit length uniformly.

Monte-Carlo Method

- Monte-Carlo method is an experimental method that aims to reach the result by trying with random numbers. In this way, it is aimed to solves mathematical and physical problems. This algorithm is also known as the Metropolis algorithm. The algorithm aims to go to predictive solutions for problems where it is difficult to find exact solutions. So it's built on probability theory.

- Monte-Carlo simulation method is a system based on probability theory. In the Monte-Carlo method, it is essential to simulate and solve an experiment or a physical event that needs to be solved by using random statistical and mathematical techniques by repeatedly using random numbers.

Monte-Carlo Method

- It is a method often used when limited computational resources are available. For example, the Monte-Carlo method was used in the Manhattan project, where the first atomic bomb was developed world war 2.

- It is a technique used to solve probabilistic or specific problems where the time factor is not important, using regular (U(0.1), random numbers in the range (0-1). Monte Carlo simulation is often used in static simulation models.

Monte-Carlo Method

- Monte-Carlo simulation is defined as a plan made by using random numbers in the solution of **deterministic** or **probabilistic** problems where the time flow is not important today. Monte-Carlo simulation, which is static rather than dynamic , is one of the best known and widely used simulation methods and is based on the assumption that parameters can be modeled with probability distributions

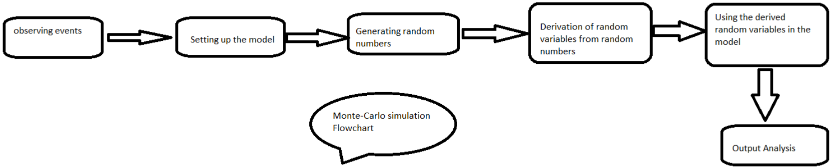


Figure: Monte-Carlo Flowchart

Monte-Carlo Example

- **Steps for finding value of π**
- Let us consider a square of 1/4 quadrant only. However the ratio of points is the same (p=m/N).

Step1: Start a random number generator.
Step2: Give the number of times shots to be fired ,say N.
Step3: Initialize the counter m, to count the random number lying within the unit circle.
Step4: Begin a loop to shoot N-times.
Step5:Fetch the random numbers for x and y such that $0 <= x <= 1$ and, 0<= y<= 1.
Step6:If $x^2 + y^2 <= 1$
the pair (x,y) lies within the quarter circle, increment the counter m=m+1.
Step7: Loop started at step 4, ends.
Step8: Calculate the value of $\pi = 4m/N = P$ [5]

Monte-Carlo Example

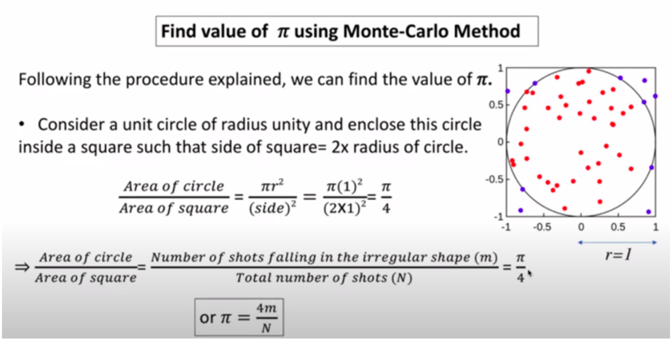


Figure: Finding value of π

Question Solution

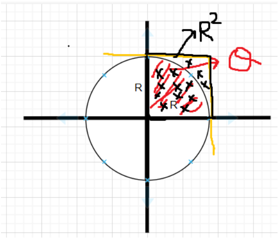


Figure: Finding π

$\pi = A/R^2$
Q = A/4

$$\frac{Q}{R^2} = \frac{area\ of\ 1/4\ circle}{area\ of\ square\ R} = \frac{n(Q)}{n(square)} = \frac{10}{13}$$

Finally ,Multiply the result by 4 according to the formula.

Discrete Methods

Discrete Methods

Discrete Methods can be used to transform uniform random variables and obtain simple distributions.
After uniform random variable obtained [0,1], it can be plugged in some simple methods to generate simple distributions:

Bernoulli Distribution

Bernoulli distribution gives outputs 0 or 1 depends on the uniform random variable (U) and probability of success (p).

$$X = \begin{cases} 1 & U < p \\ 0 & U \geq p \end{cases}$$

$$P\{success\} = P\{U \quad < \quad p\} = p.$$

Thus a Bernoulli has generated.

Binominal Distribution

Sum of n Bernoulli trial for n uniform random variable.

Example:

```
n=20;
p=0.53
U=rand(n,1);
X=sum(U<p);
```

Geometric Distribution

Count of the tries until first success Bernoulli trail.
Example:

```
X=1;
while rand > p;

  X=X+1;
end;
```

Negative Binominal Distribution

Count of the tries until rth success Bernoulli trail. It is like Geometric Distribution but stopping condition can be set.

Example:

```
X=1;
r=2;
successCount=0;
while successCount<r;

  X=X+1;

  if rand > p;

    successCount = successCount + 1;

  end;
end;
```

Arbitrary Discrete Distribution

In bernoulli distribution only success or not success situations happens but it can be extended in arbitrary discrete distribution. For example:

$$X = \begin{cases} x_0 & U < p_0 \\ x_1 & p_0 \leq U < p_1 \\ \ldots & \ldots \\ \ldots & \ldots \\ \ldots & \ldots \\ x_i & p_{i-1} \leq U \leq p_i \end{cases}$$

$$p_i = P\{X = x_i\} \quad , \quad \sum_i p_i = 1$$

Generating Discrete Variables

1. Divide interval [0,1] into sub-intervals as much as needed or infinite number of times.
2. Obtain an Uniform Random Variable from a source (From a random number generator or a table).
3. If U belongs to A_i, then $X = x_i$.

These steps are the same steps to obtain X value from Bernoulli trial, only difference [0,1] interval not divided into 2 but divided more and there is more corresponding x values other than 0 and 1.

Poisson Distribution

To generate a poisson distribution value first a lambda (λ) constant needed to use in the function.
Poisson variable takes values $x_0 = 0$, $x_1 = 1$, $x_2 = 2$, ... with probabilities

$$p_i = \mathsf{P}\{\mathsf{X} = x_i\} = e^{-\lambda}\frac{\lambda^i}{i!}$$

Now an uniform random variable U will be generated and find the set A_i containing U.

$$\mathsf{F(i\text{-}1)} \leq \mathsf{U} < \mathsf{F(i)}$$

Inverse Transform Method

Overview

Inverse transform is a method for generating random numbers from any probability distribution by using its inverse cumulative distribution $F^{-1}(x)$. The cumulative distribution for a random variable X is $F_X(x) = P(X \leq x)$.

In what follows, we assume that our computer can, on demand, generate independent realizations of a random variable U uniformly distributed on [0,1].

The Concept

- Suppose U $\sim$ Unif [0,1] and F is a one-dimensional CDF.
- Then, $F^{-1}(U) =$ X has the distribution F.
- Here we define, $F^{-1}(u) = inf\{x : F(x) \geq u\}$.

Example: Exponential Distribution

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Step 1: Calculate CDF

- $F(x) = \int_0^x \lambda e^{-\lambda x}$ when we solve this integral we get the CDF of the exponential distribution:

$$f(x) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Step 2: Calculate F^{-1}

- Solve for x: $y = 1 - e^{-\lambda x}$:
- x $= \frac{-ln(1-y)}{\lambda}$

Step 3: Plug-in the Random Variable U

- x $= F^{-1}(u) = \frac{-ln(1-u)}{\lambda}$
- This is the final solution.
- If we sample from a random variable with this distribution, we will get X that's exponentially distributed with parameter lambda.

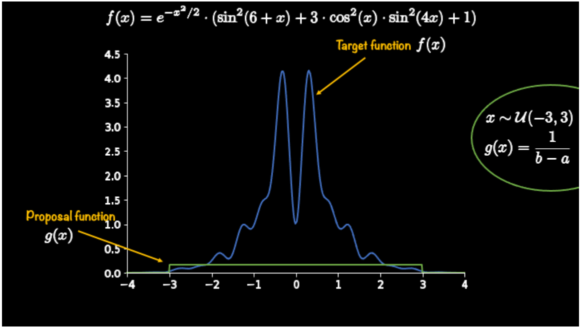
Rejection Method

What is Rejection Method?

- It is a Monte Carlo method.
- It uses randomness to generate samples.
- Main idea is generating random numbers using other distribution methods. If we get a sample which follows given distribution, then we accept it. If it doesn't follow, then we reject it.

Why Do We Need Rejection Method?

- Sometimes we don't know how to sample from a distribution(A sophisticated distribution).
- Solution is to use another, easy to sample, distribution function.
- Now we have 2 different distribution functions:
  1. $f(x)$ = Target Distribution Function
  2. $g(x)$ = Proposal Distribution Function



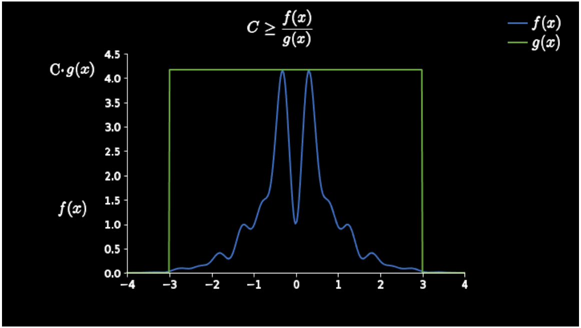
Which Distribution Function to Choose?

- Selected distribution function must be fast and easy to sample.
- Uniform Distribution is the best candidate.
- Probability density function of Uniform Distribution:

$$P(x) = 1/(b - a)$$

What is Acceptance Criterion?

- As we see with previous picture, area of proposal function doesn't cover the area of target distribution function.
- To cover, we use a constant(scaling constant) "C" and multiply proposal function with this constant.
- C is unique for every distribution function. So how do we find it?
- This is one of the problems of rejection method.

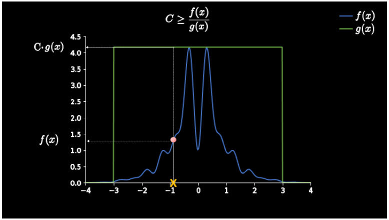


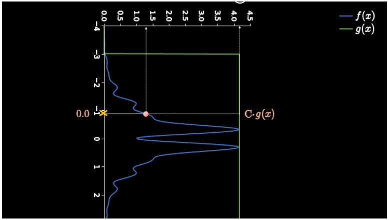
Sampling

- After getting a sample, to know if sample falls on below the target distribution function, we can use another Uniform Distribution.
- New uniform distribution function has a range between 0 and C.g(x)
- Using this new distribution function we can write our acceptance criteria as:

$$Sample \rightarrow u = \mathcal{U}(0, C.g(x))$$

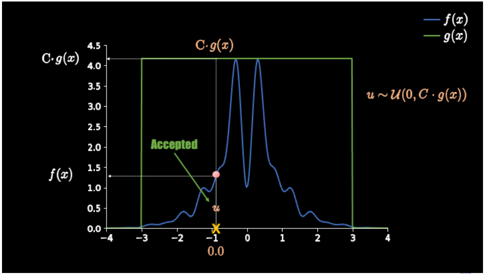
$$Acceptance\ Criteria \rightarrow u \leqslant f(x)$$





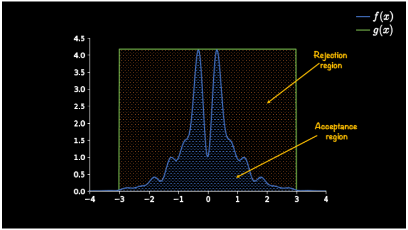
Sampling - Continue

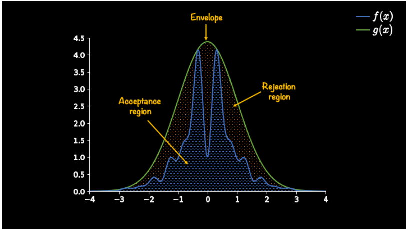
- Now it is possible to decide if a generated sample follows target distribution function or not.
- After deciding on repeat count, procedure to generate samples is as follows:
  1. Generating a random number using proposal distribution function (x)
  2. Generate a random number using Uniform distribution function (u) between 0 and C.g(x)
  3. After check for equation: $u \leqslant C.g(x)$. If it is correct, then accept it. If it is not, then reject it.
  4. Repeat the procedure until you generate enough samples.



Decreasing the Rejection Rate

- According to our target distribution function, we can select our proposal distribution function.
- Although it may decrease the rejection rate, most of the time using Uniform Distribution is better due efficiency.





[4]

Generation of Random Vectors

Introduction

A random vector is a generalization of a single random variable to many. If X and Y are random variables, than pair (X,Y) is a random vector. Distribution of the pair is called joint distribution and their individual distributions are called marginal distributions. Usually, one random variable is not enough since lots of variables are considered at once at scientific studies and this brings a need for producing random vectors instead of a random variable.

Methods For Generating Random Vectors

Rejection method can be used for generating Random Vectors. In this case, The bounding box now becomes a multidimensional cube instead of a 2D square as can be seen in the below images.

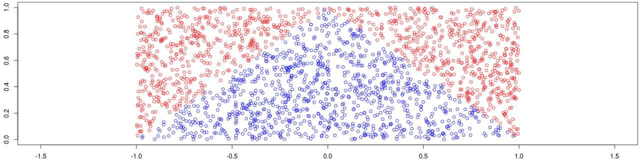


Figure: Generating Random Variables using Rejection Method.

Methods For Generating Random Vectors

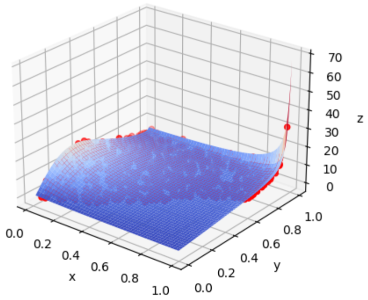


Figure: Generating Random Vectors using Rejection Method.

In Figure 5, we generate a Uniformly distributed random vector $(X_1, X_2, ..., X_n, Y)$ accepted only if $Y \leq f(X_1, ..., X_n)$. Then, the generated vector $(X_1, ..., X_n)$ has the desired joint density

$$\begin{aligned} f_{X_1,\ldots,X_n}(x_1,\ldots,x_n) &= \int f_{X_1,\ldots,X_n,Y}(x_1,\ldots,x_n,y)dy \\ &= \int_0^{f(x_1,\ldots,x_n)} 1\,dy = f(x_1,\ldots,x_n). \end{aligned}$$

Rejection method can be too inefficient even when it is used for random variables since it generates lots of variables and accepts a small portion of them. When it comes to creating random vectors, inefficiency increases significantly. So, inverse transform method is preferable if the interval is too large or the accepted area is too small.

Special Methods

Special Methods

Some random variables are generated by methods other than inverse transforms due to their complex form.

- Poisson Distribution
  Count the number of "rare events" occurring during one unit of time.
- Box-Muller Transformation
  This algorithm is created as a more economic and efficient alternative for Inverse Transform method. It converts a pair of generated Standard Uniform variables (U1, U2) into a pair of independent Standard Normal variables (Z1, Z2) using the following equation:

$$Z_1 = \sqrt{-2ln(U_1)}cos(2\pi U_2)$$

$$Z_2 = \sqrt{-2ln(U_2)}sin(2\pi U_2)$$

[2]

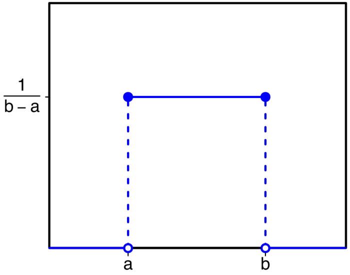


Figure: Before Box-Muller Transformation (Standart Uniform Distribution)

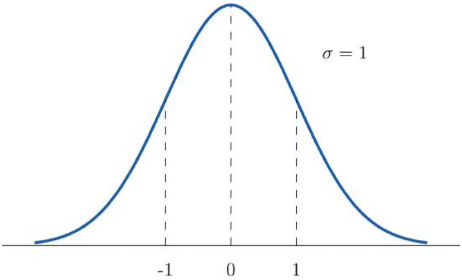


Figure: After Box-Muller Transformation (Standart Normal Distribution)

[3]

Sources

[1] Michael Baron. *Probability and Statistics for Computer Scientists*. CRC Press, 2013.

[2] H. Paul Keeler. "Simulating Poisson random variables –Direct method". In: (2015). DOI: https://hpaulkeeler.com/simulating-poisson-random-variables-direct-method/.

[3] Lu Zhu Qidi Peng Henry Schellhorn. "Generating Random Vectors Using Transformation with Multiple Roots and its Applications. (English)". In: *Applications and Applied Mathematics* 10.1 (2015), pp. 50–70. DOI: https://www.pvamu.edu/mathematics/wp-content/uploads/sites/49/04_r767-peng-vol.-10-issue-1-posted-06-18-15.pdf.

[4] Kapil Sachdeva. *What is Rejection Sampling?* URL: https://towardsdatascience.com/what-is-rejection-sampling-1f6aff92330d.

[5] scratchapixel. *Monte Carlo Methods in Practice*. URL: https://www.scratchapixel.com/lessons/mathematics-physics-for-computer-graphics/monte-carlo-methods-in-practice/generating-random-numbers.

Thank you!