# Randomness, Pseudorandomness, Randomization, Random Number Generator, Entropy

(Group 25)
Arindam, Sandeep, Aditya

Indian Institute of Information Technology, Allahabad

November 22, 2018

Today, we are going to look at one of the very important aspect of Simulation. Which will cover following topics:

## Topics we are going to cover

- What is Randomness and what is it's use?
- What is simulation and why is it needed?
- Pseudorandomness
- Random Number Generator and it's types
- Entropy and it's types

**What is Randomness?**

- Randomness is the lack of pattern or predictability in events.

- A random sequence of events, symbols or steps has no order and does not follow an intelligible pattern or combination.

- By definition, each individual random events are **unpredictable**, but in many cases the frequency of different outcomes over a large number of events (or "trials") is **predictable**.

- Randomness also is also structured. That means randomness is not absolutely random. For example, when tossing a coin, the outcome of any particular toss is unpredictable, but over a many experiments we will get equal number of heads and tails.

Many children games depends on random-output like rolling dice, spinning wheels, shuffling card, etc. all of those games rely on chances of outcomes which ideally should be random. Most important aspect about randomness is, it must be fair.

### What is about this random structure that makes it seems fair?

1. Nobody knows about the outcome before it happens.
2. All possible outcomes are equally likely.

How likely or how unlikely the event is can be explored by the help of **Simulation**.

# Simulation

**What is Simulation?**

- Simulation means imitation of a operation of a real-world process or system.
- Simulation consists of sequence of random outcomes which models a situation.
- By the help of simulation we can check the relation between theoretical probability and experimental probability.

Theoretical probability is statistically calculated on-paper probability which ideally should be equal to experimental probability but sometimes they can also vary. This difference can be found with the help simulation by using **pseudorandomness**.

Following steps should be followed while doing the simulation.

## 7 Steps for simulation

1. Identify the component to be repeated (things to be simulated).

2. Explain how to model the outcome.

3. Explain how single trial is done and how to end the particular trial.

4. Declare response variable (what you are looking at the end).

5. Fix number of trials to be performed and run trials.

6. Keep note of whats happening. Analyze the response.

7. State conclusion.

**What is Pseudorandom?**

- A pseudorandom process is a process that appears to be random but is not.

- Pseudorandom sequences typically exhibit statistical randomness while being generated by an entirely deterministic causal process.

Such a process is easier to produce than a genuinely random one, and has the benefit that it can be used again and again to produce exactly the same numbers, which is useful for testing and fixing software.

**What is Randomization?**

Randomization is the process of making something random; in various contexts this involves, for examples:

- Generating a random permutation of a sequence (such as when shuffling cards);

- Selecting a random sample of a population (important in statistical sampling);

- Allocating experimental units via random assignment to a treatment or control condition;

- Generating random numbers (see Random number generation);

- Transforming a data stream (such as when using a scrambler in telecommunications).

## Arithmetic Method

1 **Sequential**: The next random number is determined by one or several of its predecessor according to a fixed mathematical formula.

2 **The mid square method**: Start with
$Z_0 = 4$ - digit positive integer,
$Z_1$ = middle 4 digits of $(Z_0)^2$
(append 0 if necessary to left to get exactly 8 digits);
$Z_2$ = middle 4 digits of $(Z_1)^2$,
with decimal point at left and so on.

Randomness,
Pseudoran-
domness,
Randomiza-
tion, Random
Number
Generator,
Entropy

(Group 25)
Arindam,
Sandeep,
Aditya

Introduction

Randomness

Simulation

Pseudo-
randomness

Randomization

LCG

Entropy

References

# Example

Example of Mid Square Method.

## Example 1

1. $(77)^2 = 5929$
2. $(92)^2 = 8464$
3. $(46)^2 = 2116$
4. $(11)^2 = 0121$
5. $(12)^2 = 0144$
6. $(14)^2 = 0196$
   . . . . .
   . . . . .

## Example 2

1. $(73)^2 = 5329$
2. $(32)^2 = 1024$
3. $(02)^2 = 0004$
4. $(00)^2 = 0000$
5. $(00)^2 = 0000$
   . . . . .
   . . . . .

**Example 2 will create a problem as now it will generate all zeros**

**Challenges in mid square method**

- Not really random
- Entire sequence determined by $Z_0$
- If a $Z_i$ ever reappears, the entire sequence will be recycled
- If $Z_i$ becomes 0, it will generate the next random numbers as 0

## Linear Congruential Generator

- Generate a sequence of integers $Z_1, Z_2, Z_3...$via the recursion
  $Z_i = (aZ_{i-1} + c)(mod$ m$)$
- a, c and m are carefully chosen constants
- Specify a seed $Z_0$ to start off
- All the $Z_i$'s are between 0 and m-1
- Return the $i^{th}$ random number as $U_i = Z_i/m$

Randomness,
Pseudoran-
domness,
Randomiza-
tion, Random
Number
Generator,
Entropy

(Group 25)
Arindam,
Sandeep,
Aditya

Introduction

Randomness

Simulation

Pseudo-
randomness

Randomization

LCG

Entropy

References

# Example of Linear Congruential Generator

$m = 63, a = 22, c = 4, Z_0 = 19$

| Linear Congruential Generator | | | |
|---|---|---|---|
| i | $22\,Z_{i-1} + 4$ | $Z_i$ | $U_i$ |
| 0 | - | 19 | - |
| 1 | 422 | 44 | 0.6984 |
| 2 | 972 | 27 | 0.4286 |
| 3 | 598 | 31 | 0.4921 |
| 4 | 686 | 56 | 0.8889 |
| . | . | . | . |
| . | . | . | . |
| 61 | 158 | 32 | 0.5079 |
| 62 | 708 | 15 | 0.2381 |
| 63 | 334 | 19 | 0.3016 |
| 64 | 422 | 44 | 0.6984 |
| . | . | . | . |

# Linear Congruential Generator

## Random Number Generation Applications

- Gambling
- Statistical Sampling
- Computer Simulation
- Cryptography
- Data Compression

**Entropy** is defined as the average information obtained per source symbol

### The expression for Entropy

$H(x) = -\sum P(X) \log P(X_i)$, where $\log P(X_i) = I(x_i)$.

If the log in the above equation is taken to be to the base 2, then the entropy is expressed in bits. If the log is taken to be the natural log, then the entropy is expressed in nats. More commonly, entropy is expressed in bits, and unless otherwise noted, we will assume a logarithm with base 2.

Example 1. To compute the entropy of a fair coin, we rst dene its distribution

$P(X=heads)=\frac{1}{2}$

$P(X=tails)=\frac{1}{2}$

$H(x)=-\sum_{x \epsilon heads,tail} \log P(X_i)$

wehave : $H(P) = \sum P(x)logP(x) = \frac{1}{2} \log \frac{1}{2} + \frac{1}{2} \log \frac{1}{2} = 1.$

## Types of entropy

- Source Entropy
- Destination Entropy
- Joint Entropy
- Equivocation Entropy

**Source Entropy:** Source Entropy denoted by H(X) is defined as the average amount of the information obtained by the source per symbol.

**Destination entropy:** Destination Entropy denoted by H(Y) is defined by average amount of information obtained per destination symbol.

**Joint entropy:** Joint Entropy denoted by H(X,Y) is defined by average amount of information obtained per destination symbol.

**Conditional Entropy:** It is denoted by H(Y/X) is the average of information obtained at destination when it is known that X is sent.

$H(x) = - \sum_{x \epsilon Y} \sum_{\epsilon y} P(X, Y) \log P(X_i, Y_i)$

**Equivocation entropy:** Equivocation Entropy denoted by $H(X/Y)$, reverse of conditional entropy, defined as average amount of information obtained at source when it is known that $Y$ is received at destination.

$$\left\{ \begin{array}{l} H(X) + H(Y) \, when \, X,Y \, are \, independent \\ H(X) + H(\frac{X}{Y}) \, when \, X,Y \, are \, dependent \end{array} \right\}$$

The main practical application of Entropy is Data Compression. While communicating to other party we need to send large amount of data. And sometimes the capacity of channel is less in comparison with the size of data. So in this case data compression will be useful.

There are many techniques used for data compression and one of them is **HUFFMAN CODING**

# Huffman Coding

## Procedure

The Huffman coding algorithm can be summarized as follows:-

1. Think of the $p_i$ as the leaf nodes of a tree. In constructing a Huffman code by hand it's sometimes useful to sort the $p_i$ in decreasing order.

2. Starting with the leaf nodes, construct a tree as follows. Repeatedly join two nodes with the smallest probabilities to form a new node with the sum of the probabilities just joined. Assign a 0 to one branch and a 1 to the other branch. In constructing Huffman codes by hand, it's often helpful to do this assignment in a systematic way, such as always assigning 0 to the branch on the same side.

Randomness,
Pseudoran-
domness,
Randomiza-
tion, Random
Number
Generator,
Entropy

(Group 25)
Arindam,
Sandeep,
Aditya

Introduction

Randomness

Simulation

Pseudo-
randomness

Randomization

LCG

Entropy

References

### Procedure

The codeword for each symbol is given by the sequence of 0's and 1's starting from the root node and leading to the leaf node corresponding to the symbol.

Example :- Consider a source with symbols $s_1, s_2, s_3, s_4$ with probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}$ respectively.
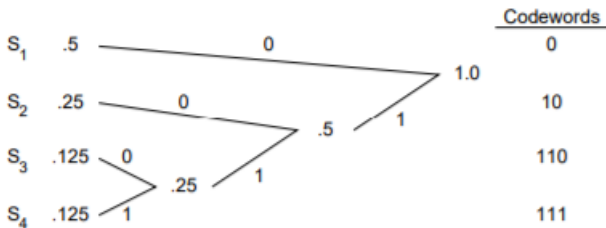
From the tree constructed, we can read off the codewords. For example, the codeword for $s_2$ is found by starting at the root node and following the branches labeled 1 then 0 to arrive at $s_2$. Therefore, the codeword for $s_2$ is 10.

The average number of bits per symbol for this code is

average length $= (1)(\frac{1}{2}) + (2)(\frac{1}{4}) + (3)(\frac{1}{8}) + (3)(\frac{1}{8})$
$= 1.75$ bits/symbol

In the example above, the rate of the Huffman code is exactly the entropy, so that the Huffman code is the best possible.

# References

📄 Marvizadeh, S. Z.
Entropy applications in industrial engineering.

📄 Pal, N. R. and Pa, S. K. (1991).
Entropy: A new definition and its applications.

📄 Ross, S. M. (2014).
*Introduction to probability models.*
Academic press.

Randomness,
Pseudoran-
domness,
Randomiza-
tion, Random
Number
Generator,
Entropy

(Group 25)
Arindam,
Sandeep,
Aditya