

II John von Neuman

- ① Initialize a random seed value
- ② Square it
- ③ Slice out middle digits

* Random number

→ number chosen by chance i.e. randomly from a set of numbers

→ Properties

- | | |
|----------------|----------------------------------|
| → random | → uniform |
| → portable | → uncorrelated |
| → efficient | → do not repeat over long period |
| → reproducible | of time |

* Pseudo random number

PRNs is a number that appears random but is actually generated using a deterministic processes.

→ follow a predictable sequence if algorithm and seed are known.

Properties

- Deterministic
- Periodic
- fast/efficient
- statistical randomness
- seed dependency

```
import random
```

```
random.seed(42)
```

```
print(random.random()) # one RN b/w 0-1
```

```
print(random.randint(1, 100)) # one RN
```

Congruential method.

→ generates a sequence of random numbers using the recurrence relation:

$$x_{n+1} = (ax_n + c) \bmod m$$

x_n = current random number

a = multiplier (a constant)

c = increment (a constant)

m = modulus (determines the range)

x_0 = seed (initial value)

* 32-bit linear congruential algorithm

$$x_n = (16807 \times x_{n-1}) \bmod (2^{31} - 1)$$

16807 → Miracle number

```
def LCA(seed, a, c, m, n):
```

```
    numbers = []
```

```
    x = seed
```

```
    for _ in range(n):
```

```
        x = (a * x + c) % m
```

```
        numbers.append(x)
```

```
    return numbers
```

seed = 32

a = 16807

c = 0

m = $2^{31} - 1$

n = 100

print(LCA(seed, a, c, m, n))

Advantages:

- simple and efficient

- easy to implement

- works well with proper parameter selection

Disadvantages:

- Predictable

- correlation in generated numbers
(not suitable for cryptography)

- Finite period (repeats)

Random variable

Consider a event with random outcomes A_1, A_2, \dots, A_k

If the event occurs $N \ggg 1$

Then,

$$\text{probability of occurrence of } A_k \quad (P_k) = \lim_{N \rightarrow \infty} \left(\frac{N_k}{N} \right)$$

N_k : no of times $\otimes A_k$ has occurred.

$$\sum P_k = 1$$

$$0 \leq P_k \leq 1$$

Eg: For die

$$P_k = 0.1667$$

$$\sum P_k = 1$$

$$\begin{aligned} \text{Expected value (mean)} &= E[x] = \sum_k x_k P_k \\ &= 1 \times 0.1667 + \dots + 6 \times 0.1667 \\ &\approx 3.5 \end{aligned}$$

$$\text{Var}(x) = E[x^2] - (E[x])^2$$

$$E[x^2] = 1^2 + 2^2 + \dots + 6^2 = 15.1667$$

$$\text{Var}(x) = 15.1667 - (3.5)^2 = 2.9167$$

Statistical error

→ the difference in measurement

The best estimate of true value

$$(\text{mean}) \quad \bar{A} = \frac{1}{n} \sum_{i=1}^n A_i; \quad n \Rightarrow \text{no of times measured} \\ A_i \Rightarrow \text{value at each measurement.}$$

$$\text{standard error} = \frac{6}{\sqrt{n}}$$

$$\text{Actual value} = \bar{A} \pm \text{Error}$$

Standard error tells how much our estimated mean could be off.

Variance tells us how much the individual measurements deviate from average

$$\text{variance } (\sigma^2) = \frac{1}{n} \sum (A_i - \bar{A})^2$$

* Relationship betwⁿ variance and standard error

$$\langle \delta A^2 \rangle = \sigma^2 \left(1 - \frac{1}{n} \right)$$

$$\text{Error} = \sigma$$

$$\sqrt{n}$$

$$= \frac{1}{\sqrt{n}} \sqrt{\frac{n}{n-1} \langle \delta A^2 \rangle}$$

$$= \frac{1}{\sqrt{n}} \sqrt{\frac{n}{n-1} \sum (A_i - \bar{A})^2}$$

$$= \sqrt{\frac{\sum (A_i - \bar{A})^2}{n(n-1)}}$$

* Monte Carlo integration (Area under curve)

$$I = \int_0^1 f(x) dx$$

Algo:

1. Picking N random points $x_1, x_2, \dots, x_n \in [0, 1]$

2. Evaluate $f(x)$ at these points

3. Average result

$$\text{i.e. } I \approx \frac{1}{N} \sum_{i=1}^N f(x_i)$$

Introducing concept of PDF $p(x)$

$$\langle F \rangle = \frac{1}{N} \sum_{i=1}^N f(x_i) p(x_i)$$

Date:

Page:

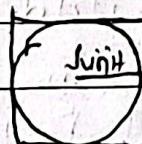
For uniform distribution where $x \in [0, 1]$

$p(x_i) = 1$, Then,

$$\mathbb{E} = \int_0^1 f(x) dx \approx \langle f \rangle \cong \frac{1}{N} \sum_{i=1}^N f(x_i)$$

$$\text{Variance} = \sigma_f^2 = \frac{1}{N} \sum_{i=1}^N (f(x_i) - \langle f \rangle)^2 p(x_i)$$

$$= \frac{1}{N} \sum_{i=1}^N f(x_i)^2 - (\langle f \rangle)^2$$



$$\frac{\pi r^2}{a^2} = \frac{\pi r^2}{2 \times 2} = \frac{\pi}{4}$$

Estimation of π using MC

Step 1: Initialize variables

$\rightarrow \text{circle_points} = 0$ (points inside the circle)

$\rightarrow \text{square_points} = 0$ (points inside the square)

$\rightarrow \text{Iterations} = 0$

Step 2: Generate random x -coordinate $\in [-1, 1]$

Step 3: Generate random y -coordinate $\in [-1, 1]$

Step 4: Check if the point is inside circle

$$d = x^2 + y^2$$

If $d \leq 1$; increment circle-points

Step 5: Increment square-points

Step 6: Repeat from step 2 for N iterations

Step 7: Estimate π

$$\pi \cong \frac{\text{circle_points}}{\text{square_points}}$$

#

Importance sampling

is a variance reduction technique in MC methods where samples are drawn from a carefully chosen probability distribution $p(x)$ instead of a uniform distribution.

→ the goal is to prioritize sampling from regions where the function $f(x)$ is large, making the estimation more efficient and accurate.

For. $I = \int f(x) dx$

$$I \approx \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)}$$

where x_i are sampled from $p(x)$.

→ Detailed: Instead of evaluating

$$I = \int_a^b f(y) dy$$

we rewrite it using our chosen probability distribution $p(y)$:

$$I = \int_a^b \frac{F(y)}{p(y)} p(y) dy$$

using our change of variable $y(x)$

$$I = \int_a^b \frac{F(y(x))}{p(y(x))} dx$$

Using MC, we approximate integral with

$$\bar{I} \approx \frac{1}{N} \sum_{i=1}^N \frac{F(y(x_i))}{p(y(x_i))}$$

This approach ensures that we sample more points where the function is important, making the estimate more accurate with fewer samples.

Monte-Carlo acceptance and rejection

simple & effective way to generate random numbers ~~without~~ from arbitrary probability distribution function without requiring complex transformation or integration.

Concept

Suppose we want to sample from probability distribution function $p(x)$ over an interval $x \in [a, b]$

Let M be the maximum value of $p(x)$ such that

$$M \geq p(x) \quad \forall x \in [a, b]$$

Generate two random numbers,

$\rightarrow x$ from ^{uniform distribution} $f(x)$ over $[a, b]$

$\rightarrow s$ from uniform distribution over $[0, M]$

Accept x if $p(x) \geq s$, otherwise reject it and reiterate.

Advantages

- No need for complex transformation or integration
- Works simply by rejecting samples that do not fit the desired probability distribution.
- The process is straightforward.

Example:

Consider a function

$$f(x) = 1.2 - x^2 \quad x \in [0, 1]$$

1. The integral of $f(x)$ is 1 making it a valid probability function

2. The minimum value of $f(x)$ is 1.2 at $x=0$

Algorithm:

1. Pick 2 random nos:
 x from $(0,1)$
 y from $(0,1.2)$

2. If $y \leq f(x)$ accept x
3. Otherwise, reject and repeat.

Markov chain

→ special type of stochastic process defined as a collection of random quantities $\{X_t\}_{t \in T}$ for some set T .

→ Used to describe the systems where next state depends only on the present state not on past states. (Markov property)

$$P(\theta^{(n+1)} | \theta^{(n)}, \theta^{(n-1)}, \dots, \theta^0) = P(\theta^{(n+1)} | \theta^{(n)})$$

→ State-space

set of all possible states a system can be in:

Eg: coin toss $\Rightarrow \{\text{head}, \text{tail}\}$

rolling die $\Rightarrow \{1, 2, \dots, 6\}$

→ Transition probability,

probability of moving from one state to another.

$p(x,y)$ is probability of moving from x to y .

→ Homogeneous markov chains

If the transition probabilities do not change over time, the Markov chain is called homogeneous.

Eg: if a website visitor has 30% chance of clicking an ad. today, the same probability holds tomorrow.

Eg:

States: {sunny, rainy, cloudy}

Transition probabilities:

If today is sunny, tomorrow has:

→ 70% chance of sunny

→ 20% chance of cloudy

→ 10% chance of rainy.

It follows markov chain because tomorrow's weather depends only on today's

Transition matrix

$$P = \begin{bmatrix} P(S \rightarrow S) & P(S \rightarrow R) & P(S \rightarrow C) \\ P(R \rightarrow S) & P(R \rightarrow R) & P(R \rightarrow C) \\ P(C \rightarrow S) & P(C \rightarrow R) & P(C \rightarrow C) \end{bmatrix}$$

$$= \begin{bmatrix} 0.7 & 0.1 & 0.2 \\ 0.3 & 0.5 & 0.2 \\ 0.4 & 0.2 & 0.4 \end{bmatrix}$$

3 Random Walk

is a process where a particle or object moves step-by-step in different directions based on probability. The future position depends only on the current position, making it a markov chain.

Example : Drunkard's walk

Imagine a drunk person standing at a starting point.

→ with probability p he moves 1 step forward (+1)

" " " " " backward (-1)

or he stays in the same position (0)

Each step is random and depends only on his current position.

This is 1D random walk.

Transition probabilities in random walk:

→ Move right (↗) : $P(x \rightarrow x+1) = p$

→ " left (↖) : $P(x \rightarrow x-1) = q$

→ " " " : $P(x \rightarrow x) = r$

→ No other moves are possible

where $p = 0.4$ (40% chance of moving right)

$$q = 0.4$$

$$r = 0.2$$

$$p + q + r = 1$$

If the particle is at point 0, after 1 step possible outcomes are:

→ position 1 with probability 0.4

" -1 " " 0.4
still at 0 with probability 0.2

Properties of RW

(+) Displacement after n steps = 0 (Averaged)

→ if you take many walks, the average position remains close to the starting point.

→ but for a single walk you might end up far from where you started.

o

(ii) Root mean squared deviation increases.

→ it measures how far your site typically are from the start after many steps.

Ehrenfest Model

(Context: Modelling gas molecules using a Markov chain)

Imagine a box divided into two equal parts by a partition, with a small hole in the middle. Inside the box, there are N gas molecules that can move randomly between the two boxes.

To model this we assume:

- each molecule moves randomly from one side to another
- the movement of each molecule is independent of others.
- at a moment, a single is chosen in random

This set up is known as Ehrenfest model, which helps us to understand how gas molecules distribute themselves over a time.

Strongest model as markov chain

Let's define $X(n)$ as a random variable as the no. of molecules in one half of the box after n steps.

Since the no of molecules is fixed ' N ', if there are x -molecules in one box there are $N-x$ molecules in other half.

State space $\Rightarrow \{0, 1, 2, \dots, N\}$

The transition probability below describes how the no of molecules in one half changes and is given by:

$$P(x,y) = \begin{cases} x/N, & \text{if } y=x+1 \text{ (a molecule moves to 1st box)} \\ 1-x/N, & \text{if } y=x-1 \text{ (", " " " out of 1st box)} \\ 0, & \text{if } |y-x| \neq 1 \text{ (any other transition)} \end{cases}$$

where,

x = no. of molecules in 1st box

N = total no of molecules

Conclusion

④ Random motion leads to equilibrium

Over time each half = $N/2$

→ This models the idea of thermal equilibrium in physics.

(ii) Markov property

→ The no of molecule at next step only depend on current state but no on how it got there.