

* Discuss MC method to estimate value of π .

Ans: Monte Carlo method provides a stochastic approach to estimate the value of π using random sampling.

Concept:

First of all, we consider a square of length 1 spanning from $(0,0)$ to $(1,1)$. Then, a quarter circle of radius 1 & having center at $(0,0)$ is inscribed.

The area of square is (1×1) & area of quarter circle is $\frac{\pi \times 1^2}{4}$

If we randomly generate points (x,y) within the square, the proportion of points that lands inside quarter circle approximate $\pi/4$

$$\text{i.e. } \frac{\pi}{4} \approx \frac{M}{N} \quad \text{Where, } M = \text{no. of points inside quarter circle}$$

Then, we estimate the value of π

$$\text{as, } \pi \approx 4 \times \frac{M}{N}$$

$N = \text{Total no. of points generated}$

Algorithm:

Algorithm:

- I Generate N random points (x, y) within the unit square where $x, y \in [0, 1]$
- II Check if each point satisfies $x^2 + y^2 \leq 1$ (inside the quarter circle)
- III Count the number of points M that falls inside the circle.
- IV Compute π using the formula
$$\pi \approx 4 \times \frac{M}{N}$$
- V Increase N for better accuracy.

Code:

$N = 10000$

$M = 0$

~~for~~ iter = 0 to

 for iter in range(1, N+1):

$x, y = \text{random.uniform}(0, 1),$

$\text{random.uniform}(0, 1)$

 if $x**2 + y**2 \leq 1;$

 iter = iter + 1

 pi_estimate = $4 * (M/N)$

print f"Estimated pi value: {pi_estimate}"

* Algorithm to evaluate simple integral using Monte Carlo Method:

The Monte Carlo method is useful for evaluating definite integrals when the integral is difficult to compute analytically. It relies on random sampling to approximate the area under a curve.

Problem statement:

Evaluate the integral

$$\int_a^b f(x) dx$$

Algorithm:

- ① Define the integral: Identify the function $f(x)$ to integrate & the limits a & b .
- ② Generate N random points x_i uniformly distributed over the interval $[a, b]$ by using formula
$$x_i = a + (b - a) \times \text{random}(0, 1)$$
- ③ Calculate the function value at each random point $f(x_i)$
- ④ Compute the mean of the function values:

$$\text{Mean} = \frac{1}{N} \sum_{i=1}^N f(x_i)$$

⑤ Multiply the average by the interval length to get the integral approximation:

$$I \approx (b-a) \times \text{Mean}$$

* How do you evaluate

$$I = \iint_{-5}^5 dx dy g(x,y) \text{ with}$$

$$g(x,y) = \exp(-x^2 - y^2 - (x-y)^2/2) \text{ using}$$

Monte Carlo. Illustrate with algorithm.

Ans: Monte Carlo relies on random sampling to estimate the integral

Algorithm:

① Define the integral: The functionals to integrate is

$$g(x,y) = \exp(-x^2 - y^2 - (x-y)^2/2) \text{ if } \\ \text{The limit of integration is } [-5,5] \\ x, y \in [-5,5]$$

② Generate N random pairs (x_i, y_i) , uniformly distributed over interval $x, y \in [-5, 5]$ by using formula

$$x_i = -5 + (5 - (-5)) \times \text{random}(0, 1)$$
$$y_i = -5 + (5 - (-5)) \times \text{random}(0, 1)$$

- (III) Evaluate the functional value at each random pairs.

$$f_i = g(x_i, y_i)$$

- (IV) Compute the mean/average of the functional values

$$\text{Mean} = \frac{1}{N} \sum_{i=1}^n f_i$$

- (V) Multiply the average of by the interval width to get the integral approximation.

$$I \approx (5 - (-5)) (5 - (-5)) \times \text{Mean}$$

$$I \approx 100 \times \text{Mean}$$

- (VI) Increase the value of N to get more accurate integral value.

* Illustrate the importance sampling to evaluate integral of a function.

Ans: Importance Sampling is a variance reduction technique used in Monte Carlo methods to estimate integrals more efficiently. It works by sampling from a distribution $p(x)$ that is more 'important' or has higher weight in regions where the integrand is significant.

Problem statement:

To evaluate the integral of a function:

$$I = \int_a^b f(x) dx$$

Direct MC integration might not be efficient if $f(x)$ has sharp peaks or is highly variable. Importance sampling improves efficiency by introducing a proposal distribution $p(x)$ that resembles the shape of $f(x)$ & rewriting the integral as

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

Algorithm:

- ① Choose a distribution $p(x)$ that is similar to the shape of $f(x)$, easy to sample from & satisfies $p(x) > 0$ wherever $f(x)$ is non zero.
- ② ~~Draw~~ Generate N random samples x_i from one distribution $p(x)$.
- ③ for each random sample x_i , compute the weight value:
$$w_i = \frac{f(x_i)}{p(x_i)}$$
- ④ Compute integ estimated integral as:

$$I \approx \frac{1}{N} \sum_{i=1}^N w_i$$

e.g. Evaluate the integral:

$$I = \int_0^1 \frac{1}{1+x^2} dx$$

- ① Choose a uniform distribution $[0, 1]$.
 $p(x) = 1$ for $x \in [0, 1]$
- ② Generate ~~Re~~ random N random numbers samples from $p(x)$

iv) for each value random sample x_i , compute the weighted value

$$w_i = \frac{f(x_i)}{p(x_i)} = \frac{1}{1+x_i^2} = \frac{1}{1+x_i^2}$$

v) Compute estimate integral as:

$$I \approx \frac{1}{N} \sum_{i=1}^N \frac{1}{1+x_i^2}$$

* What is the significance of importance sampling over simple sampling? Illustrate considering one example to evaluate integration of a function.

Ans: Importance Sampling is a variance reduction technique in Monte Carlo methods that improves efficiency by strategically sampling from regions of high impact.

The significance of importance sampling over simple sampling are:

① Variance reduction:

- Simple sampling estimates integrals by averaging uniform samples, leading to high variance.

Importance sampling

Importance sampling chooses a probability distribution that resembles the function, reducing fluctuations in estimates.

① Variance reduction:

Simple sampling estimates integrals by averaging uniform samples, leading to high variance if the integrand varies widely (rare event regime)

Importance Sampling biases sampling towards 'important' regions (peaks or tail) & weights result to correct bias. This reduces variance significantly

② Computational Efficiency:

Requires fewer samples to achieve the same precision as simple sampling, especially for high dimensional integrals or skewed distributions

Handling Rare Events:

→ Efficiently sample low-probability regimes (e.g. financial risk modelling for extreme losses).

* How do you distinguish 'Stationary distribution' from other distributions? Explain algorithm to get 'Stationary distribution' using python code.

Ans: A stationary distribution of a Markov chain one is the probability distribution that remains unchanged as the chain evolves over time. If π is the stationary distribution, then if it is in the stationary distribution at one time step, it will remain in that distribution at all future time steps.

If $\pi = [\pi_1, \pi_2, \dots, \pi_n]$ is the stationary distribution with transition matrix P ,

$$\pi P = \pi$$

A stationary distribution π of a Markov chain is unique in two key ways:

Invariance:

If subspecies

$\pi = \pi P$, where, P is one transition matrix. Unlike transient distribution, it remains unchanged over time steps

$$\pi(x) P(x, y) = \pi(y)$$

Irreducible: The chain can move between any pair of states, ensuring entire state space is explored.

Aperiodic: The chain does not stuck in cycle, allowing convergence.

2. Long term Behaviour:

For irreducible & aperiodic chains, π represents the limiting distribution i.e. $\lim_{n \rightarrow \infty} \pi^n = \pi$ i.e. $\lim_{n \rightarrow \infty} P^n = \pi$
i.e. $\lim_{n \rightarrow \infty} P^n(x, y) = \pi(y)$

Whereas other distributions depends on initial condition.

3. Uniqueness:

Unlike arbitrary probability distributions stationary distributions are unique under ergodicity (i.e. irreducible + aperiodic chains).

* Algorithm to Compute Stationary Distribution:

Efficient way to compute the stationary distribution is through Power iteration, which repeatedly multiplies the transition matrix by an initial guess until converge once.

Algorithm Steps:

1 Initialization:

Start by initializing a probability vector π_0 of the size n , where n is the no. of states in the Markov chain.

The initial distribution can be chosen as a uniform distribution (i.e. each state has an equal prob. of $\frac{1}{n}$) or randomly generated while ensuring the probability sum is 1.

Code:

```
import numpy as np
# No. of states
n = 3
# Initialize a uniform distribution
pi = np.ones(n)/n
print ("Step 1: initial Distribution  
Uniform") ; pi
```

2: Matrix multiplication:

Multiply the current probability vector π_k by the transition matrix P to obtain the next vector.

$$\pi_{k+1} = \pi_k \cdot P$$

This step advances the distribution to the next state based on transition probabilities.

Transition matrix

$$P = \text{np.array}\left(\left[\begin{array}{ccc} 0.5, 0.3, 0.2 \\ 0.1, 0.6, 0.3 \\ 0.4, 0.2, 0.4 \end{array}\right]\right)$$

Multiply the current distribution vector

by the transition matrix

$$\text{new_pi} = \text{np.dot}(\pi, P)$$

print

3. Normalization:

To maintain a valid probability distribution, normalize the result to ensure the sum of prob. is 1.

$$\pi_{k+1} = \frac{\pi_k}{\sum \pi_{k+1}}$$

Code:

$$\text{new_pi} = \text{new_pi} / \text{np.sum}(\text{new_pi})$$

4. Convergence Check:

Check the difference between the current & previous distribution vectors using a distance metric (like the L2 norm).

$$\|\pi_{k+1} - \pi_k\| < \text{tolerance}$$

If the difference is below a chosen threshold (like 10^{-8}), the algorithm has converged.

Code:

```
tolerance = 10-8
difference = np.linalg.norm(new_pi - pi)
if tolerance < difference:
    print("Final stationary distribution:", new_pi)
else:
    iteration = iteration + 1
```

① Repeat:

Repeat the step multiplication & normalization steps until convergence is achieved or maximum no. of iteration is reached.

Initialize iteration

iteration = 0

max_iteration = 1000

if iteration < max_iteration & tolerance > difference
or

report the step 2, 3, 4
else: iteration = iteration + 1;

print("The stationary distribution:",
new_pi)

* Discuss the role of Master Equation in Monte Carlo Methods (8 marks)

Ans: The Master Equation describes the temporal change in the probability distribution $P(x, t)$ of a system being in state x at time t . It is commonly written as:

$$\frac{dP(x, t)}{dt} = \sum_{x'} [W(x' \rightarrow x) P(x', t) - W(x \rightarrow x') P(x, t)]$$

Where,

$P(x, t)$ = Probability of being in state x at time t

$W(x' \rightarrow x)$ = Transition rate from state x' to state x .

The first term in the right hand side represents the gain of probability from other states x' to x .

The second term in the right hand side represents the loss of probability from state x to other states x' .

Roles in Monte Carlo Methods :

i) It determines how the system transitions from one state to another state over time.

ii) It achieves detailed balance

$\tau_i W(i \rightarrow j) = \tau_j W(j \rightarrow i)$ to sample from stationary distributions.

- ③ The master equation helps simulate complex system by guiding how random samples are drawn to replicate the underlying probabilistic transitions.
- ④ Monte Carlo simulations uses the Master's Equation to update state probabilities iteratively, making it fundamental for dynamics simulation & modeling time-dependent processes.
- ⑤ By understanding the transition rates, efficient algorithms (like Metropolis-Hastings, Gibbs sampling) can be used to design to reach the stationary distribution more efficiently.

* Explain the significance of 'Reversible Markov Chain.'

Ans: A reversible Markov Chain is a Markov chain that satisfies the detailed balance condition, which ensures that the chain behaves the same in forward & backward directions when it reaches its stationary distribution.

A markov chain with transition matrix P & stationary distribution π is called reversible if it satisfies the following condition for all states i, j

$$\pi(i) \cdot P(i \rightarrow j) = \pi(j) \cdot P(j \rightarrow i)$$

$\pi(i)$ = Stationary probability of state i

$P(i \rightarrow j)$ = Transition probability from state i to j

The significance of 'Reversible Markov chain' are:

① Reversible Markov chains guarantee that the stationary distribution is maintained. This property ensures that the samples are drawn from the current distribution.

② A reversible Markov chain satisfies the detailed balance condition:

$$\pi_i P_{ij} = \pi_j P_{ji}$$

This ensures that the transition's bet? states do not drift away from the equilibrium.

This results in faster convergences & reduced computation effort.

- ④ The reversibility property makes the Markov chain time-reversible

$$P_{ij}^{(r)} = \frac{\pi_j}{\pi_i} P_{ji} = P_{ji}$$

which enables backward simulation in Markov chain

- ⑤ Reversible chains avoid biases in sampling which is critical for Bayesian inference & high-dimensional integration & allow efficient sampling.

- ⑥ Reversible chains have real eigenvalues for their transition matrices which makes the convergence faster.

- ⑦ Reversible Markov chains make the simulation robust & consistent, even in complex systems.

* Explain the Meaning of "Continuous State Space"

Ans: A state space S is said to be 'continuous' if it consists of an uncountably infinite set of states, typically represented by as a subset of the real numbers (\mathbb{R}) or higher dimensional spaces (\mathbb{R}^n).

Eg: A state space defined over the interval $[0, 1]$.

A multidimensional space like \mathbb{R}^2 for two variables.

Since the number of states is uncountable, we describe the probability distribution using a probability density function (PDF). The pdf $f(x)$ satisfies:

$$P(a \leq x \leq b) = \int_a^b f(x) dx$$

Eg: The set of all possible values that a Gaussian random variable can take forms a continuous state space over the entire real line \mathbb{R} .

In Metropolis-Hastings or Gibbs Sampling, the state space can be continuous, as the algorithms generate samples from continuous probability distributions.

Roles in Monte Carlo Methods:

- ① Many MC methods involve generating samples from continuous distributions such as normal or exponential distribution.
- ② Continuous state spaces are vital in MCMC when sampling from posterior distributions in Bayesian statistics.
e.g. In Gibbs Sampling, each variable can take continuous values & the sampling process must take account for these continuous transitions.
- ③ MC integration often involves integrating functions over continuous domains which requires sampling from continuous state spaces.

Continuous state spaces are essential when modeling systems with real valued or high dimensional parameters. They are fundamental in various MC methods, especially in fields requiring accurate sampling & integration. Understanding and working with continuous state space enables precise simulation & probabilistic modeling in complex applications.

* Write down the meaning of transition probability with examples

Ans: Transition probabilities define the likelihood of moving between states in Markov Chain.

For states i & j , the transition probability p_{ij} is the conditional probability:

$$p_{ij} = P(X_{t+1} = j / X_t = i)$$

which is the probability of transitioning from state i to state j .

Suppose for a 3-state chain containing states Bull market, Bear market & Stagnant market with Transition matrix i.e. matrix containing transition probabilities:

$$\text{Transition matrix } (P) = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 0.6 & 0.2 & 0.2 \\ 2 & 0.2 & 0.5 & 0.3 \\ 3 & 0.3 & 0.1 & 0.6 \end{bmatrix}$$

1: Bull market

2: Bear market

3: Stagnant market

Then, from the first row, (if we have state as bull)

Bull - Bull \rightarrow There is 60% chance of remaining bullish

Bull - Bear \rightarrow There is 20% chance of remaining going to bear market from Bullish

Bull - Stagnant \rightarrow There is 20% chance of going to transitioning from state Bull to Stagnant

* What is state space? Explain

Ans: The state space of a stochastic process is the set of all possible states that the process can occupy.

The state space represents all the potential values that a random variable can ~~assume~~^{take} during the simulation. It is crucial for defining the range over which random sampling is performed.

In Markov Chain MC methods, the state space encompasses all the possible configurations that the Markov Chain can visit during its simulation.

For instance, in random walk, the state space represents all the possible positions that a particle or walker can occupy at each step.

When estimating the value of π using the MC integration method, the state space consists of all the points (x, y) within the square of side length 2 centered at origin. Points inside the unit circle are counted to estimate π .

Therefore, the state space in MC methods is not just a set of values but a comprehensive representation of all potential configurations that the simulation explores.

* What is a random walk? Explain

Ans: In context of Markov Chain Monte Carlo (MCMC), a random walk is a stochastic process where an entity moves through a state space step by step, with each step being determined probabilistically based only on their current state. This process exhibits the Markov property, meaning that the future state depends solely on the present state & not on the sequence of states that preceded it.

e.g Consider a 2D random walk on a grid where a point (x, y) moves one step at a time. At each step, the point can move in one of four directions: up, down, left, right. The transition probabilities are equal, i.e.

- $P(\text{up}) = P(\text{down}) = P(\text{left}) = P(\text{right}) = 0.25$.

If current position is $(3, 4)$, the next position could be one of the following:

Up : $(3, 5)$

Down : $(3, 3)$

Left : $(2, 4)$

Right : $(4, 4)$

