TP 547- Princípios de Simulação de Sistemas de Comunicação

Prof. Samuel Baraldi Mafra

Inatel
Instituto Nacional de Telecomunicações

Monte Carlo

- Em 1946 o matemático Stanislaw Ulam durante um jogo de paciência tentou calcular as probabilidades de sucesso de uma determinada jogada utilizando a tradicional análise combinatória;
- Após gastar bastante tempo fazendo cálculos percebeu que uma alternativa mais prática seria simplesmente realizar inúmeras jogadas e contar quantas vezes cada resultado ocorria;
- Stanislaw Ulam comentou com seu amigo John von Neumann sobre o experimento e que queria utilizar o ENIAC;
- Uso de métodos de amostragem estatística para solucionar o problema da difusão de nêutrons em material sujeito a fissão nuclear, difundindo assim sua aplicação.

- O nome deve-se ao cassino de Monte Carlo do principado de Monaco, onde o tio de Stanislaw Ulam jogava constantemente;
- Método de Monte Carlo pode ser descrito como método de simulação estatística que utiliza sequencias de números aleatórios para desenvolver simulações;

Frequência relativa

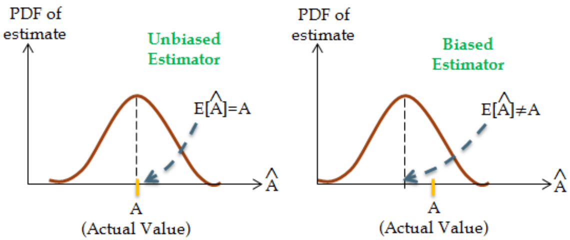
- A estimativa de Monte Carlo é baseada na interpretação da frequência relativa da probabilidade. Ao definir a frequência relativa, a primeira etapa é especificar um experimento e um evento de interesse, A.
- A probabilidade do evento A é aproximada pela frequência relativa do evento, que é definida por N_A/N. A probabilidade do evento A, definida no sentido da frequência relativa, é obtida replicando o experimento aleatório um número infinito de vezes.

Estimadores

Estimadores consistentes e sem viés

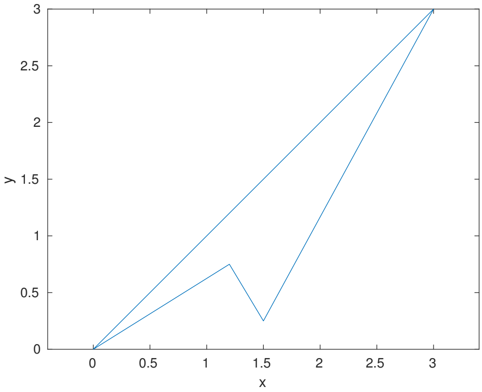
Os estimadores devem ser consistentes, ou seja, quando o número de amostras cresce, a resposta converge para a probabilidade esperada.

Os estimadores devem ter viés zero (unbiased), ou seja, para uma grande quantidade de amostras a média das amostras deve ser igual ao valor real da probabilidade esperada.



Calculo de Área

Calcular o valor da área da região abaixo:



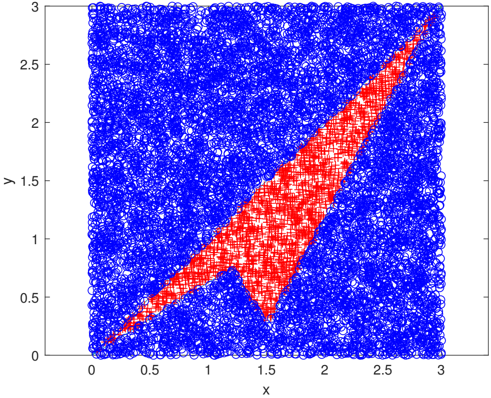
xv=[0,1,3,1.5,1.2,0];
yv=[0,1,3,0.25,0.75,0];

Calculo de Área

- Gerar pontos uniformemente distribuídos em uma área quadrada que contenha a região;
- Verificar o número de pontos que fica dentro da região: Usar a função inpolygon do Matlab ou a função path da biblioteca matplotlib
- Calcular a seguinte relação:

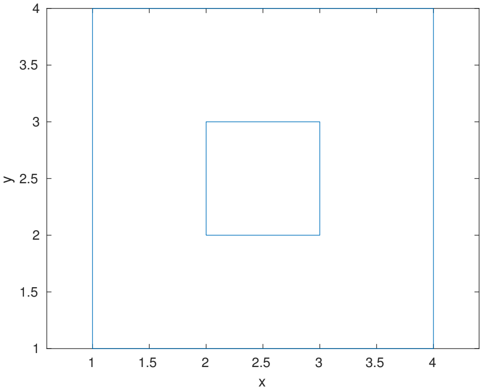
$$\frac{A_{Regiao}}{A_{quadrado}} = \frac{N_{regiao}}{N_{quadrado}}$$

```
import numpy as np
from matplotlib import path
import matplotlib.pyplot as plt
N = 100000

# Define the polygon
p = path.Path([(0, 0), (1, 1), (3, 3), (1.5, 0.25), (1.2, 0.75), (0, 0)])

# Generate random points
x = np.random.uniform(0, 1, size=N)
y = np.random.uniform(0, 1, size=N)

# Scale the points
xab = 3 * x
yab = 3 * y

# Check if the points are inside the polygon (vectorized)
Pb = p.contains_points(np.stack((xab, yab), axis=1))

# Count the points inside the polygon
count = np.sum(Pb)

# Calculate the area
area = 9 * count / N

# Print the area
print(area)
```

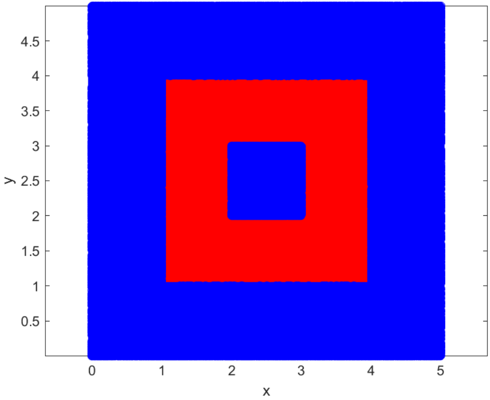

3
2.5
2
1.5
1
0.5
0
y
0
0.5
1
1.5
2
2.5
3
x

Calculo de Área

Calcular o valor da área da região entre os dois quadrados abaixo:



```python
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import path

# Define the outer and inner squares
outer_square = path.Path([(1,1), (4, 1), (4, 4),(1,4), (1,1)])
inner_square = path.Path([(2, 2), (3, 2), (3, 3), (2, 3), (2, 2)])

# Generate random points inside the outer square
N = 1000000
x = np.random.uniform(0, 5, N)
y = np.random.uniform(0, 5, N)

# Check if points are inside the outer square but not the inner square
inside_outer = outer_square.contains_points(np.stack((x, y), axis=1))
inside_inner = inner_square.contains_points(np.stack((x, y), axis=1))
count = np.sum(np.logical_and(inside_outer, np.logical_not(inside_inner)))

# Print the count
print(f"Number of points inside the outer square but not the inner square: {count}")

# Calculate the area
area=25*count/N
print(area)
```

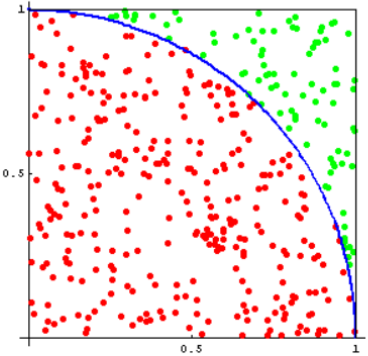


4.5
4
3.5
3
y
2.5
2
1.5
1
0.5
0
1
2
3
4
5
x

Como calcular o valor do Pi usando Monte Carlo?

Calcular usando simulação de Monte Carlo o valor de π.

- Área do quadrado é dada por $Aq = L^2 = 1$;
- Área da região circular é dada por $Ac = 1/4 * \pi * r^2 = 1/4 * \pi$
- $Ac/Aq = Nc/Nq = 1/4 * \pi$
- $\pi = 4 * Nc/Nq$



```
import numpy as np
import matplotlib.pyplot as plt

N = 1000000

# Generate random numbers in one go
x = np.random.uniform(size=N)
y = np.random.uniform(size=N)

# Calculate squares of x and y in one go
x2 = x ** 2
y2 = y ** 2

# Calculate distance from origin in one go
dxy = np.sqrt(x2 + y2)

# Find points within the circle in one go
inside_circle = dxy <= 1

# Count the number of points inside the circle
counts = np.sum(inside_circle)

# Calculate the estimate for pi
piestimation = 4 * counts / N
# Print the result
print("O valor de pi e", piestimation)
```

- Seja $g : \mathbb{R} \longrightarrow \mathbb{R}$ uma função real de variável real e considere o problema de calcular a seguinte integral:

$$I = \int_0^1 g(x)\mathrm{d}x.$$

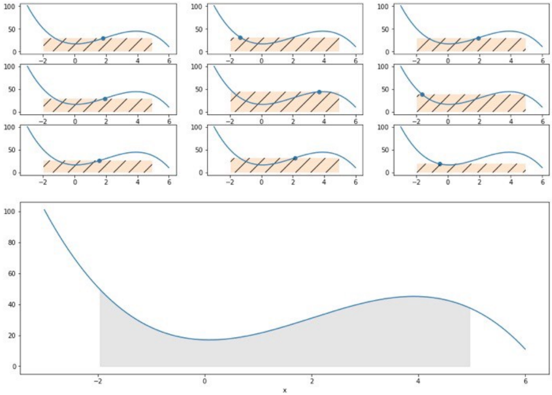
- Considere que U seja uniformemente distribuído com função densidade de probabilidade $f_U = 1 \quad 0 < x < 1.$
- O valor esperado de $g(U)$ é dado por:

$$\mathbb{E}[g(x)] = \int_0^1 g(x) f_U \mathrm{d}x = \int_0^1 g(x) = I$$

Podemos aproximar I a partir da geração de uma grande quantidade de números aleatórios $U_1, U_2, \ldots, U_n$ com distribuição uniforme em $[0,1]$ e então, calcular a aproximação como sendo a média:

$$I \approx \sum_{i=0}^{n} \frac{g(U_i)}{n}.$$

Média de áreas de retângulos



Integração de Monte Carlo

Algoritmo

Gerar n números aleatórios $U_1, U_2, \ldots, U_n$ uniformemente distribuídos em $[0,1]$.
Calcular $g(U_1), g(U_2), \ldots, g(U_n)$.
Calcular a média desses valores.

Integração de Monte Carlo: Exemplo

Calcule a seguinte integral:

$$I = \int_0^1 e^{-\frac{x^2}{2}} \mathrm{d}x$$

O valor da integral é 0.855624.

```python
import numpy as np
import matplotlib.pyplot as plt

N=100000
x=np.random.uniform(0,1,N)
integral=sum(np.exp((-x**2)/2))/N
print(integral)
```

Para o caso mais geral:

$$I = \int_a^b g(x)\mathrm{d}x.$$

Realiza-se a substituição de variáveis:

$$y = \frac{x-a}{b-a} \qquad \mathrm{d}y = \frac{\mathrm{d}x}{(b-a)}.$$

$$I = \int_a^b g(x)\mathrm{d}x = \int_0^1 g((b-a)y+a)(b-a)\mathrm{d}y$$

Algoritmo

- Gerar n números aleatórios $U_1, U_2, ..., U_n$ uniformemente distribuídos em $[0,1]$.
- Gerar n números aleatórios com distribuição uniforme entre a e b
- Calcular a função g para estes números gerados.
- Calcular a média desses valores.
- Multiplicar por (b-a)

Integração de Monte Carlo: Exemplo

Calcule a seguinte integral:

$$I = \int_3^8 x^2 e^{-\frac{x^2}{2}} \mathrm{d}x$$

O valor da integral é 0.003671.

```
import numpy as np
import matplotlib.pyplot as plt

N=1000000
y=np.random.uniform(0,1,N)
integral=5*sum(pow((5*y+3),2)*np.exp((-(5*y+3)**2)/2))/N
print(integral)
```

Para o caso com limites $a = 0$ e $b = \infty$:

$$I = \int_0^\infty g(x)\mathrm{d}x.$$

Realiza-se a substituição de variáveis:

$$y = \frac{1}{1+x} \qquad \mathrm{d}y = -\frac{\mathrm{d}x}{(1+x)^2} = -y^2 dx.$$

$$I = \int_0^\infty g(x)\mathrm{d}x = -\int_1^0 g(1/y-1)/y^2\mathrm{d}y = \int_0^1 g(1/y-1)/y^2\mathrm{d}y$$

Calcule a seguinte integral:

$$I = \int_0^{\infty} e^{-x/2} \mathrm{d}x$$

O valor da integral é 2.

```python
import numpy as np
import matplotlib.pyplot as plt

N=1000000
y=np.random.uniform(0,1,N)
integral=1*sum(np.exp((-(1/y-1)/2))/y**2)/N
print(integral)
```