

Trabalho Probabilidade - Monte Carlo

Gustavo Pamplona Melo Coelho

7 de março de 2024

1 Introdução

O método Monte Carlo é uma técnica estatística e computacional que utiliza a aleatoriedade para resolver problemas diversos, especialmente em áreas como física, estatística, finanças, engenharia, entre outras. O método leva o nome do famoso cassino Monte Carlo, conhecido por seus jogos de azar e aleatoriedade.

1.1 História

O método Monte Carlo teve início durante a Segunda Guerra Mundial, quando cientistas como Stanislaw Ulam e Nicholas Metropolis, envolvidos no Projeto Manhattan, buscavam maneiras de simular e entender o comportamento de nêutrons em reatores nucleares. O nome "Monte Carlo" foi sugerido por Ulam, inspirado pelos jogos de azar realizados na cidade de Monte Carlo.

2 Aplicação

2.1 Aproximação do valor de π usando o método Monte Carlo

O código (Figura 1) implementa uma simulação de Monte Carlo para estimar o valor de π . Inicialmente, são gerados 10 milhões de pontos aleatórios (x,y) uniformemente distribuídos no quadrado unitário (0,1). Em seguida, é realizado um loop que percorre cada ponto gerado, verificando se ele está localizado dentro do primeiro quadrante do círculo unitário (centrado na origem). Para isso, é utilizado o critério $x[i]^2 + y[i]^2 < 1$. Se o ponto está dentro do círculo, um contador é incrementado. Ao final, o valor de π é estimado multiplicando a razão de pontos dentro do círculo pelo total de pontos gerados, e multiplicando por 4, já que a área de um quadrante do círculo é $\pi/4$. O resultado é impresso (Figura 2), fornecendo uma aproximação de π utilizando métodos estatísticos baseados em números aleatórios.

2.2 Verificar a probabilidade de sair cara no lançamento de uma moeda utilizando o método de Monte Carlo

O código (Figura 3) implementa O método Monte Carlo para calcular a probabilidade de sair cara no lançamento de uma moeda. Envolve a geração de números aleatórios para simular o processo em questão e, em seguida, calcular a proporção de vezes em que o evento desejado ocorre.

Para provar a probabilidade de sair cara em um lançamento de moeda justa usando o método de Monte Carlo, os passos seguidos foram:

1. Geração de Números Aleatórios: Gere um grande número (N) de números aleatórios entre 0 e 1. Cada número pode ser interpretado como uma simulação de um lançamento de moeda, onde valores menores que 0,5 representam coroa e valores maiores representam cara.
2. Contagem de Caras: Conte quantos dos números gerados estão acima de 0,5.
3. Cálculo da Probabilidade Estimada: A probabilidade estimada de sair cara será dada pela razão entre o número de caras e o número total de simulações (N).

O resultado (Figura 4) convergirá para 0,5 à medida que você aumenta o número de simulações (N). Isso reflete a probabilidade teórica de 0,5 para uma moeda justa.

```

1 N <- 1000000 # Número de iterações
2 n <- 0
3
4 # Gerando as coordenadas dos pontos no quadrado unitário
5 #segundo uma distribuição uniforme entre 0 e 1.
6 x <- runif(N)
7 y <- runif(N)
8
9 # Percorrendo as listas geradas de x e y, se o ponto gerado estiver
10 #dentro do primeiro quadrante do círculo, somamos 1 no contador n.
11 for (i in 1:N) {
12   if (x[i]^2 + y[i]^2 < 1) {
13     n <- n + 1
14   }
15 }
16
17 # Como a área de um quadrante do círculo é pi/4, multiplicamos a razão n/N por 4.
18 pi_estimado <- 4 * n / N
19
20 pi_estimado

```

Figura 1: Código estimando valor de pi

```

Saída do programa

[1] 3.140112

[Execution complete with exit code 0]

```

Figura 2: Resultado código estimando π

2.3 A probabilidade de sair o número 2 no lançamento do dado é $1/6$ como prevê a definição ?

```
1 monte_carlo <- function(num_simulacoes) {  
2   # Gerando N números aleatórios entre 0 e 1 (representando lançamentos de moeda)  
3   n <- runif(num_simulacoes)  
4  
5   # Contador de quantos são maiores que 0,5 (representando caras)  
6   contador <- sum(n > 0.5)  
7  
8   # Calculando a probabilidade estimada  
9   probabilidade <- contador / num_simulacoes  
10  
11   return(probabilidade)  
12 }  
13  
14 # Número de simulações  
15 num_simulacoes <- 1000000  
16  
17 # Executa o método de Monte Carlo  
18 probabilidade_estimada <- monte_carlo(num_simulacoes)  
19  
20 probabilidade_estimada
```

Figura 3: Código calculando a probabilidade de sair cara no lançamento de uma moeda

```
Saída do programa  
  
[1] 0.500959  
  
[Execution complete with exit code 0]
```

Figura 4: Resultado do cálculo da probabilidade de sair cara no lançamento de uma moeda