Tutorial Métodos de Montecarlo Para Calcular el Área

El método de Montecarlo es una técnica numérica que utiliza números aleatorios para aproximar soluciones de problemas matemáticos y estadísticos. Uno de sus usos más comunes es estimar áreas de figuras geométricas, como círculos, elipses o regiones irregulares. En este tutorial, exploraremos cómo implementar el método de Montecarlo en Python para calcular áreas y proporcionaremos ejemplos prácticos.

1 Introducción al Método de Montecarlo

El método de Montecarlo se basa en el principio de generar puntos aleatorios dentro de un espacio conocido y utilizar la proporción de puntos que caen dentro de una región de interés para estimar su área. Cuantos más puntos se generen, más precisa será la estimación.

Los métodos de Montecarlo se basan en la analogía entre probabilidad y volumen. Las matemáticas de las medidas formalizan la noción intuitiva de probabilidad, asociando un evento con un conjunto de resultados y definiendo que la probabilidad del evento será el volumen o medida relativa del universo de posibles resultados. Montecarlo usa esta identidad a la inversa, calculando el volumen de un conjunto interpretando el volumen como una probabilidad. En el caso más simple, esto significa muestrear aleatoriamente un universo de resultados posibles y tomar la fracción de muestras aleatorias que caen en un conjunto dado como una estimación del volumen del conjunto. La ley de grandes números asegura que esta estimación converja al valor correcto a medida que aumenta el número de muestras. El teorema del límite central proporciona información sobre la magnitud del probable error en la estimación después de un número finito de muestras. En esencia podemos decir que el método de Montecarlo consiste en calcular o aproximar ciertas expresiones a través de adivinarlas con la ayuda de dibujar una cantidad normalmente grande de números aleatorios.

2 Aplicación en Física

El método de Monte Carlo se aplica ampliamente en física para resolver problemas complejos que a menudo no tienen soluciones analíticas sencillas o incluso conocidas. Algunas áreas donde el método de Monte Carlo es esencial incluyen:

- **Simulaciones de Sistemas Físicos:** En física, se pueden simular sistemas complejos, como interacciones de partículas en un gas, comportamiento de fluidos, o la evolución de sistemas cuánticos, utilizando el método de Monte Carlo. Las simulaciones pueden proporcionar información valiosa sobre el comportamiento de estos sistemas en diferentes condiciones.

- **Integración Numérica:** En muchos problemas físicos, es necesario calcular integrales complicadas que no tienen soluciones analíticas directas. El método de Monte Carlo se puede utilizar para estimar estas integrales numéricamente.

- **Dinámica Molecular:** En el estudio de la dinámica de moléculas y átomos, el método de Monte Carlo se aplica para simular el comportamiento y las interacciones de estas partículas en sistemas complejos, como proteínas o sólidos.

- **Física Estadística y Termodinámica:** El método de Monte Carlo es fundamental en la física estadística para el cálculo de propiedades termodinámicas de sistemas, como la energía libre, la entropía y la capacidad calorífica.

2.1 Cálculo de Áreas

Para implementar el método de Montecarlo en Python, sigue estos pasos:

1. Genera pares de coordenadas aleatorias dentro de un rango que abarque la figura de interés.

2. Evalúa si cada par de coordenadas está dentro de la figura utilizando una función que describa la forma.

3. Calcula la proporción de puntos que caen dentro de la figura con respecto al total de puntos generados.

4. Multiplica esta proporción por el área del espacio en el que se generaron los puntos para estimar el área de la figura.

Ejemplo: Cálculo del área de un círculo

Supongamos que queremos calcular el área de un círculo de radio "r" utilizando el método de Montecarlo. El círculo está inscrito en un cuadrado de lado "$2r$", lo que significa que su área es πr^2. Aquí está la implementación en Python:

```
import math
import random

def montecarlo_area_circulo(r, num_puntos):
    dentro_circulo = 0

    for _ in range(num_puntos):
        x = random.uniform(-r, r)
        y = random.uniform(-r, r)
        distancia = math.sqrt(x**2 + y**2)
        if distancia <= r:
            dentro_circulo += 1

    area_cuadrado = (2 * r)**2
    area_estimada_circulo = (dentro_circulo / num_puntos) *
   area_cuadrado
    return area_estimada_circulo

radio = 5.0
num_puntos = 100000
area_estimada = montecarlo_area_circulo(radio, num_puntos)
print(f"Area estimada del circulo: {area_estimada}")
```

Este código genera puntos aleatorios dentro del cuadrado, verifica si cada punto está dentro del círculo y, finalmente, estima el área del círculo.

Ejemplo 2: Cálculo del Área de una Región Irregular

Supongamos que queremos calcular el área de la región definida por la ecuación $y = x^2$ en el intervalo $[-1, 1]$. Aquí está la implementación:

```
import math
import random

def montecarlo_area_region_irregular(num_puntos):
    dentro_region = 0

    for _ in range(num_puntos):
        x = random.uniform(-1, 1)
        y = random.uniform(0, 1)
        if y <= x**2:
            dentro_region += 1

    area_cuadrado = 2  # Area del cuadrado en el intervalo
    [-1, 1] con altura 1
    area_estimada_region = (dentro_region / num_puntos) *
    area_cuadrado
    return area_estimada_region

num_puntos = 100000
area_estimada_irregular = montecarlo_area_region_irregular(
    num_puntos)
print(f"Area estimada de la region irregular: {
    area_estimada_irregular}")
```

2.2 Integrales Definidas con Montecarlo

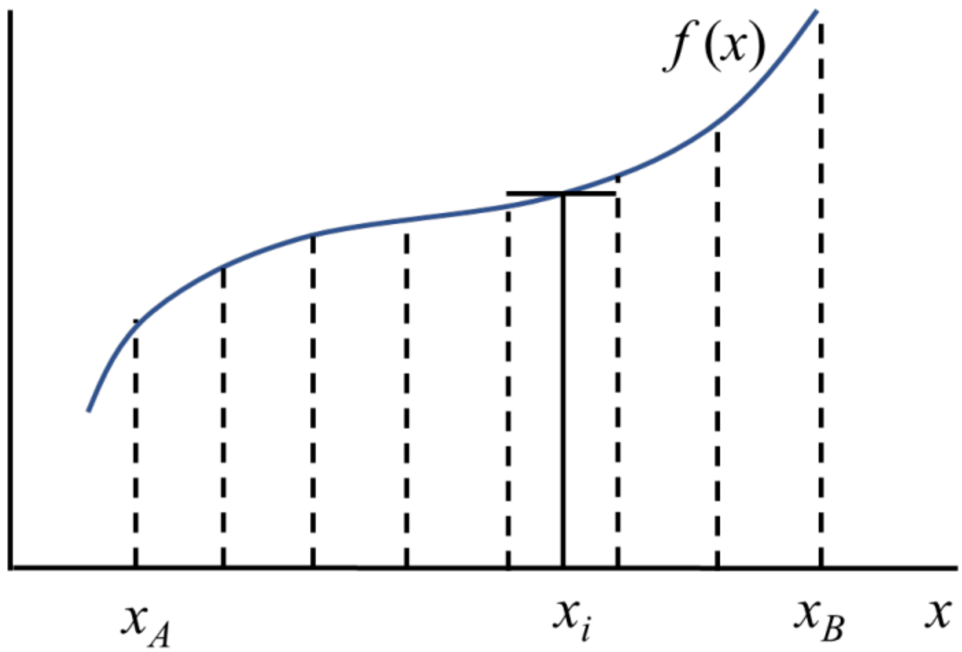
Si tenemos una función genérica $f(x)$ y queremos calcular la integral definida

$$\int_{x_a}^{x_b} f(x)dx$$

Podemos dividir el espacio en N intervalos de tamaño h

$$h = \frac{x_b - x_a}{N}$$

Y luego aproximar la íntegral como la suma de N evaluaciones $f_i = f(x_i)$ de la función f:



$$\int_{x_a}^{x_b} f(x)dx \approx \sum_{i=1}^{N} f_i h$$

Mientras mayor sea N, mejor será la aproximación.

Ejemplo: Cálculo Integral definida

Queremos calcular la integral definida:

$$I = \int_0^{\pi} sen(x)dx$$

```
import random
import math

def f(x):
    return math.sin(x)

N = 1000
xa = 0
```

```python
xb = math.pi
h = (xb - xa)/N
integral = 0

for i in range(N):
    x = random.uniform(xa, xb)
    integral += f(x)

integral *= h
print(f"integral : {integral:.3f}")
```

Integrales Múltiples

Este método se puede generalizar para calcular integrales en más de una dimensión, la diferencia es que en vez de un intervalo h tendremos un área, volumen, o hipervolumen.

Sea la integral:

$$I = \int_{\sigma} f(\vec{x}) d\vec{x}$$

Con N puntos podemos calcular el hipervolumen H

$$H = \frac{1}{N} \int_{\sigma} d\vec{x}$$

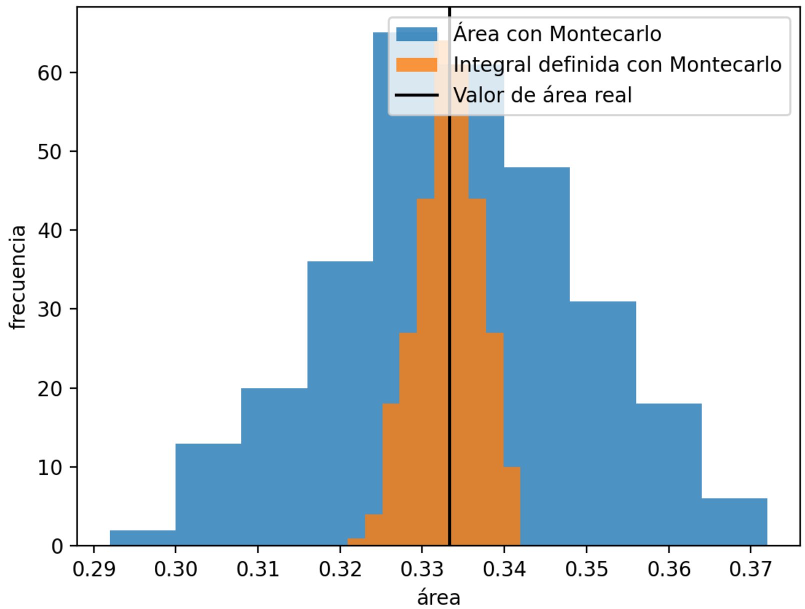
y por último:

$$\int_{\sigma} f(\vec{x}) d\vec{x} \approx H \sum_{i=1}^{N} f_i$$

Integral vs Área

Siempre que sea posible será mejor calcular un área como si fuese el cálculo de una integral definida.

En el siguiente histograma pueden ver los resultados de haber calculado una misma área con $N = 1000$ puntos.



Lo que vemos en el gráfico es que hay mucha menos dispersión cuando calculamos el área usando el método de la integral definida.

3 Conclusión

En este tutorial, aprendimos cómo implementar el método de Montecarlo en Python para calcular áreas de figuras geométricas. Hemos explorado ejemplos prácticos de cálculo de áreas de un círculo y una región irregular. El método de Montecarlo es una herramienta poderosa para aproximar soluciones numéricas y se puede aplicar en una amplia variedad de problemas matemáticos y científicos.

4 Ejercicios

1. Escribir un código en python para determinar el área de un triángulo equilátero de lado 1 usando el método de Montecarlo. Graficar los puntos dentro y fuera del triángulo.

2. Escribir un código en python utilizando el método de Montecarlo para estimar el área determinada entre dos funciones: $y = x^2$, $y = \sqrt{x}$ en el intervalo $[0, 1]$. Graficar

3. Determinar el área de una elipse mediante el método de Montecarlo. Graficar.
   La ecuación de una elipse es:

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1,$$

   donde (h, k) es el centro de la elipse, a es el semieje mayor y b el semieje menor.

4. Evaluar la integral, usando el método de Montecarlo.

$$\int_0^{\pi/2} \int_0^{\pi} sen\ u\ cos(u - \pi) dudv$$