

ALGORITMOS PROBABILISTAS Y DE MONTE CARLO

1. Algoritmos Probabilistas

Definición y Características

Los algoritmos probabilistas utilizan números aleatorios en su lógica de ejecución. A diferencia de los algoritmos deterministas, que producen siempre el mismo resultado para una entrada dada, los algoritmos probabilistas pueden generar diferentes salidas con distintas probabilidades.

Clasificación:

- Algoritmos de Las Vegas: Siempre devuelven la respuesta correcta, pero su tiempo de ejecución es aleatorio. Ejemplo: QuickSort aleatorizado.
- Algoritmos de Monte Carlo: Tienen un tiempo de ejecución fijo, pero pueden devolver una respuesta incorrecta con una pequeña probabilidad. Ejemplo: Verificación de primalidad.

Ventajas:

- Soluciones más rápidas en problemas complejos.
- Sencillez en algunos casos donde los deterministas serían complicados.
- Flexibilidad en aplicaciones con entradas grandes o con alta incertidumbre.

Desventajas:

- No garantizan exactitud absoluta.
- Requieren métodos para controlar el error probabilístico.
- Dificultad para depurar en algunos contextos debido a la aleatoriedad.

Aplicaciones Comunes:

- Criptografía: Generación de claves seguras.
- Optimización: Algoritmos genéticos y simulación por recocido.
- Análisis de datos: Algoritmos aleatorizados para muestreo o reducción de dimensionalidad.

Ejemplo Resuelto:

Problema: Encontrar un número repetido en un arreglo de 10 millones de elementos usando muestreo aleatorio.

Solución:

1. Tomar muestras aleatorias del arreglo.
2. Verificar si el número ya se ha visto antes.
3. Si se repite, detener el algoritmo.

Ventaja: Rapidez al no recorrer todo el arreglo.

Otro Ejemplo: Determinar si un número es primo usando el Test de Fermat (Monte Carlo).

```
import random

def es_posiblemente_primo(n, k=5):
    for _ in range(k):
        a = random.randint(2, n-2)
        if pow(a, n-1, n) != 1:
            return False
    return True

print(es_posiblemente_primo(97))
```

2. Algoritmos de Monte Carlo

Definición:

Los algoritmos de Monte Carlo son un subtipo de algoritmos probabilistas que usan muestreo aleatorio para obtener una solución aproximada a un problema, especialmente útiles cuando los modelos deterministas son costosos o imposibles de aplicar.

Características:

- No garantizan precisión absoluta.
- Se puede ajustar la precisión aumentando el número de simulaciones.
- Tienen aplicaciones en física, estadística, economía, inteligencia artificial, etc.

Ejemplo Clásico: Aproximación de π

Paso 1: Generar puntos (x, y) aleatorios en el cuadrado de lado 1.

Paso 2: Verificar si caen dentro del círculo de radio 1.

Paso 3: Calcular π como:

$$\pi \approx 4 \times (\text{N}^\circ \text{ puntos dentro del círculo} / \text{N}^\circ \text{ total de puntos})$$

Código en Python:

```
import random

N = 1000000

inside = 0

for _ in range(N):
    x, y = random.random(), random.random()
    if x**2 + y**2 <= 1:
        inside += 1

pi = 4 * inside / N

print(pi)
```

Ejercicio Resuelto: Integración Monte Carlo

Problema: Estimar el área bajo la curva $f(x) = x^2$ en $[0, 1]$.

Solución:

1. Generar N puntos aleatorios en el cuadrado $[0,1] \times [0,1]$.
2. Contar cuántos están bajo la curva.
3. El área aproximada es $(N^\circ \text{ puntos bajo curva} / N) * \text{área del cuadrado}$.

Código Python:

```
N = 100000

under = 0

for _ in range(N):
    x, y = random.random(), random.random()
    if y <= x**2:
        under += 1

area = under / N

print(area)
```

Más aplicaciones de Monte Carlo

- Finanzas: Valoración de opciones financieras y evaluación de riesgo.
- Física Nuclear: Modelado de trayectorias de partículas.
- Inteligencia Artificial: Algoritmos como MCTS (Monte Carlo Tree Search) usados en juegos como Go y ajedrez.
- Biología Computacional: Predicción de estructuras proteicas.

Datos Curiosos:

- El método de Monte Carlo se usó por primera vez durante el Proyecto Manhattan para simular procesos de reacción nuclear.
- Se llama "Monte Carlo" por el famoso casino, haciendo alusión al azar.
- Muchos juegos de azar se pueden modelar usando algoritmos de este tipo.
- Las simulaciones de Monte Carlo han sido usadas para estimar el número de errores de memoria RAM en condiciones espaciales.

Conclusión:

Los algoritmos probabilistas, y en especial los de Monte Carlo, ofrecen una forma poderosa de abordar problemas complejos mediante simulaciones y aleatoriedad. Aunque sacrifican precisión, su eficiencia y facilidad de implementación los convierten en herramientas indispensables en la ciencia moderna.

Con el avance de la computación paralela y los recursos distribuidos, los métodos probabilísticos seguirán creciendo en popularidad y efectividad, especialmente en campos donde la incertidumbre es parte inherente del problema.