Óscar Gómez Borzdynski

Práctica 1

Procesos Estocásticos

In [2]:
```
import random
import math
import numpy as np
```

In [2]:
```
%%html
<style>
    h1, h2, h3 {color: #3d6f91;}
    b, li, ul {color: #5D8AA8;}
</style>
```

Ejercicio 1

Se dibuja un cuadrado unitario (de lado 1) con un círculo unitario (de diámetro 1) inscrito. Se generan puntos aleatorios con distribución uniforme dentro del cuadrado.

1. ¿Cuál es el universo de eventos de este experimentos?

El universo de eventos es: $\Omega = \{X1 = \text{Caer dentro del círculo}, X2 = \text{Caer fuera del círculo}\}$

1. ¿Cual es la probabilidad que un punto generado esté incluido dentro del círcuo?

$$P(X1) = \frac{\text{Área círculo}}{\text{Área cuadrado}} = \frac{\pi}{4}$$

1. Usar la respuesta del punto 2. para escribir un programa que calcule π con tres cifras significativas. Entregar el codigo (es corto...), el valor conseguido (primera cinco cifras decimales), explicar el criterio de parada que se ha usado para determinar cuando terminar el calculo (sin usar el valor de π, claro...) y el numero de puntos que se han generado.

Sabemos que colocando puntos aleatoriamente en el cuadrado tenemos una probabiliad de caer dentro del círculo de $\frac{\pi}{4}$. Por tanto podemos calcular $\pi = 4P(X1)$.

Como criterio de parada utilizamos una tolerancia (`tol`) entre el valor de la anterior iteración y la actual. En caso de no cumplir con el criterio de parada se establece un número máximo de iteraciones (`max_iters`). Cada iteración generará un número determinado de puntos (`iter_size`)

In [3]:
```
# Constants
tol = 1e-8
max_iters = 1e6
iter_size = 1e2

# Variables
inside = 0
total = 0
previous = -1

random.seed(123)

# Code
for i in range(int(max_iters)):
    for _ in range(int(iter_size)):
        total += 1
        # Generate x and y between -0.5 and 0.5
        x = (random.random() - 0.5)**2
        y = (random.random() - 0.5)**2
        if math.sqrt(x+y) < 0.5:
            inside += 1
    actual = inside/total
    if abs(actual - previous) < tol:
        break
    previous = actual

pi = 4*actual
print(f"Número de iteraciones: {i}")
print(f"Número de puntos: {total}")
print("Valor de pi obtenido: {0:.5f}".format(pi))
print("Error: {0:.5f}".format(abs(pi - np.pi)))
```

```
Número de iteraciones: 452642
Número de puntos: 45264300
Valor de pi obtenido: 3.14189
Error: 0.00030
```

Ejercicio 2

Una caja contiene r pelotas rojas y b pelotas azules, $r \geq 1$ and $b \geq 3$. Se sacan tres pelotas de la caja (sin remplazar las que se han sacado). Usando el concepto de probabilidad condicional, calcular la probabilidad que las tres pelotas salgan en secuencia azul, rojo, azul.

$$P(pb, pr, pb) = P(pb|pr, pb)P(pr|pb)P(pb) = \frac{b}{b+r}\frac{r}{b+r-1}\frac{b-1}{b+r-2}$$

Ejercicio 3

Como ejercicio de probabilidad, este es muy fácil, pero es muy gracioso, me gusta, y quiero preguntarlo. Se trata del juego de las tres puertas.

Hay tres puertas, y detras de una de ellas hay un premio. El presentador os pide elegir una, sin abrirla. Después, abre una de las otras dos, y os muestra que allí no está el premio. Finalmente os pide si queréis cambiar la puerta o confirmar la elección inicial.

¿Cual es la decisión que maximiza la probabilidad de ganar, y qué probabilidad de ganar os da?

Sea A = El concursante elige la puerta donde está el premio

Sea B = El concursante elige la puerta donde no está el premio

Sea G = El jugador gana el premio

$$P(A) = \frac{1}{3}$$

$$P(B) = \frac{2}{3}$$

$$P(G) = P(AG) + P(BG) = P(G|A)P(A) + P(G|B)P(B)$$

Supongamos que cambiamos de puerta:

$P(G|A) = 0$ ya que hemos dejado la puerta con el premio y $P(G|B) = 1$ ya que cogemos la puerta co nel premio. Por tanto $P(G) = \frac{2}{3}$

Supongamos que no cambiamos de puerta:

$P(G|A) = 1$ ya que mantenemos la puerta con el premio y $P(G|B) = 0$ ya que dejamos la puerta con el premio. Por tanto $P(G) = \frac{1}{3}$

Con estos resultados vemos que lo más adecuado es cambiar de puerta, independientemente de la puerta inicial que cojamos.

Ejercicio 4

Dada la función $f(x) = \frac{\pi}{2}\cos(\pi x)$ si $|x| < \frac{1}{2}, 0$ resto. ¿Es una densidad de probabilidad?

Para que una función sea densidad de probabilidad, su integral en todo el dominio debe ser 1 y además debe ser no negativa en todo el dominio.

$f(x) \geq 0 \; \forall x \in \mathbb{R}$

$$\int_{-\infty}^{\infty} f(x) = \int_{-1/2}^{1/2} f(x) = \int_{-1/2}^{1/2} \frac{\pi}{2}\cos(\pi x) = \frac{1}{2}\int_{-1/2}^{1/2} \pi\cos(\pi x) = \frac{1}{2}[\sin(\pi x)]_{-1/2}^{1/2}$$

Por tanto sí vemos que es una función de densidad.

Ejercicio 5

Dada una cadena de Markov con un conjunto finito de estados $\{1, \ldots, M\}$, sea $p_m(t)$ la probabilidad que la cadena esté en el estado m al instante t. Se define el vector $p(t) = [p_1(t), \ldots, p_M(t)]'$

1. Usar la definición de la matriz de transición P para demostrar que $p'(t+1) = p'(t)P$ y por tanto, si λ es la distribución inicial $p'(t+1) = \lambda P$

Podemos definir P donde P_{ij} es la probabilidad de pasar del estado i al j. Definamos la operación $p'(t+1) = p'(t)P$, se puede demostrar que $p_m(t+1) = \sum_{i=1}^{M} P_{im}(t)p_i(t)$, es decir, es la probabilidad de llegar desde cualquier estado al estado m.

Sea $\lambda = p(0)$ el estado inicial, $p'(t) = p'(t-1)P = p'(t-2)P^2 = \cdots = p'(0)P^t$

Considerando la cadena de Markov aportada. Responder

1. Considerando $\lambda = [1, 0, 0, 0, 0, 0]'$, calcular $p(2)$, $p(10)$ y $p(100)$

In [4]:
```
P = np.array([[0.5 , 0.5 , 0   , 0   , 0   , 0   ],
              [0   , 0.5 , 0.5 , 0   , 0   , 0   ],
              [0.25, 0   , 0.25, 0.25, 0.25, 0   ],
              [0.25, 0   , 0.5 , 0.25, 0   , 0   ],
              [0   , 0   , 0   , 0   , 0.5 , 0.5 ],
              [0   , 0   , 0   , 0.25, 0.5 , 0.25]])

lam = np.array([1, 0, 0, 0, 0, 0])
```

In [5]:
```
def Markov(P, lam, niter):
    P = np.linalg.matrix_power(P, niter)
    return lam@P
```

In [6]:
```
print(Markov(P, lam, 2))
print(Markov(P, lam, 10))
print(Markov(P, lam, 100))
```

```
[0.25 0.5  0.25 0.   0.   0.  ]
[0.16009521 0.17276764 0.19525528 0.11034393 0.22227859 0.1
3925934]
[0.14285714 0.14285714 0.17142857 0.11428571 0.25714286 0.1
7142857]
```

1. Repetir el cálculo de $p(2), p(10)$ y $p(100)$ con $\lambda = [0, 0, 1, 0, 0, 0]'$

In [7]:
```
lam = np.array([0, 0, 1, 0, 0, 0])
print(Markov(P, lam, 2))
print(Markov(P, lam, 10))
print(Markov(P, lam, 100))
```

```
[0.25   0.125  0.1875 0.125  0.1875 0.125 ]
[0.14878464 0.15279961 0.17943859 0.11302376 0.24539089 0.1
6056252]
[0.14285714 0.14285714 0.17142857 0.11428571 0.25714286 0.1
7142857]
```

1. Repetir el cálculo de $p(2), p(10)$ y $p(100)$ con $\lambda = [0, 0, 0, 0, 0, 1]'$

In [8]:
```
lam = np.array([0, 0, 0, 0, 0, 1])
print(Markov(P, lam, 2))
print(Markov(P, lam, 10))
print(Markov(P, lam, 100))
```

```
[0.0625 0.     0.125  0.125  0.375  0.3125]
[0.13236713 0.12480164 0.15695381 0.11663246 0.27828407 0.1
9096088]
[0.14285714 0.14285714 0.17142857 0.11428571 0.25714286 0.1
7142857]
```

1. ¿Cómo cambian los tres vectores? ¿Cómo cambia su dependencia de las condiciones iniciales? ¿Qué implica esto para $p(\infty)$?

Podemos ver que todos los $p(100)$ son idénticos independientemente del estado inicial en el que se encuentre el sistema. Por ello, suponemos que a partir de cierto instante t_0 podemos considerar un comportamiento estable del sistema. Con todo,
$p(\infty) = [0.14285714, 0.14285714, 0.17142857, 0.11428571, 0.25714286, 0.17142$

In [ ]: