Cadena de Ehrenfest

En [1] se encuentra la siguiente definición:

Sean A y B dos urnas dentro de las cuales se encuentran distribuidas un total de N bolas de acuerdo a cierta configuración inicial, por ejemplo, la urna A tiene i bolas y en la urna B hay $N - i$ bolas. En cada unidad de tiempo se escoge una bola al azar y se cambia de urna. Para tal efecto puede considerarse que las bolas se encuentran numeradas y que se escoge un número al azar, se busca la bola con ese número y se cambia de urna.

Sea X_n el número de bolas en la urna A al tiempo n, entonces la colección $\{X_n : n = 0,1,\ldots\}$ constituye una cadena de Markov con espacio de estados finito $\{0,1,\ldots,N\}$. Este modelo fue propuesto por Ehrenfest para describir el intercambio aleatorio de moléculas en dos regiones separadas por una membrana porosa.

Ejercicios

En este ejercicio analizamos por medio de la simulación el modelo de la urna de Ehrenfest. Considere a N como el número de bolas totales.

1. Escriba todos los estados de la Cadena de Markov.

   El espacio de estados será desde tener N bolas hasta llegar 0 de la urna A y viceversa, esto se puede denotar como:

   $$\{0,1,2,\ldots,N-1,N\}$$

   Si suponemos que tenemos N=10 bolas, entonces nuestro diagrama de estado puede verse de esta forma:

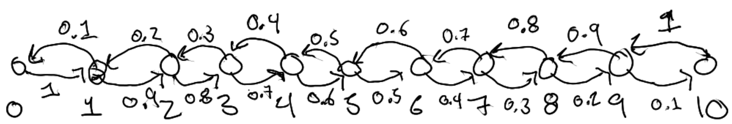

Fig. 1 – *Diagrama de estados para $N = 10$*

Generalizando para cualquier numero de bolas, podemos que decir que la

$$P(\text{Tener una bola menos de la urna A}|\text{Tenemos k bolas en la urna A}) = \frac{k}{N}$$

$$P(\text{Tener una bola más de la urna A}|\text{Tenemos k bolas en la urna A}) = \frac{N-k}{N}$$

En nuestro diagrama de estados se veria de esta forma:

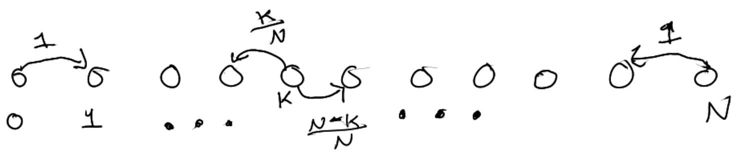

FIG. 2 – *Diagrama de estados en general*

2. Escriba la matriz de transición (y explique claramente de dónde viene la expresión). Viendo nuestro diagrama de estados anteriormente podemos observar que en cada renglon de nuestra matriz de transición tenemos dos posibilidades, una en la que hemos ganado un bola en la urna A con probabilidad de $\frac{N-k}{N}$ y otra en la hemos perdido una bola con probabilidad de $\frac{k}{N}$, ademas tenemos los casos donde tenemos cero bolas y por lo tanto la unica opción posible es que ganemos una con una probabilidad de 1 y de forma analoga cuando tenemos 50 tenemos sola posibilidad de perder una con una probabilidad de 1. Con esto en mente podemos visualizar nuestra matriz de transición de esta forma:

	0	1	2	3	4	5	...	N
0	0	1	0	0	0	0	...	0
1	$\frac{K}{N}$	0	$\frac{N-K}{N}$	0	0	0	...	0
2	0	$\frac{K}{N}$	0	$\frac{N-K}{N}$	0	0	...	0
3	0	0	$\frac{K}{N}$	0	$\frac{N-K}{N}$	0	...	0
4	0	0	0	$\frac{K}{N}$	0	$\frac{N-K}{N}$	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	$\frac{K}{N}$	0	$\frac{N-K}{N}$
N	0	0	0	...	...	...	1	0

FIG. 3 – *Matriz de transición*

3. Haga un programa que simule y grafique la trayectoria del proceso: Empieza inicialmente con una distribución inicial π_0, y simula la trayectoria X_0,X_1,...X_n hasta un tiempo n dado por el usuario.
   Para la simulación utilizamos una binomial de un ensayo con una probabilidad $\frac{j}{N}$ para conocer de esta manera si ganamos una bola o perdimos una bola, ya que nuestra función nos regresas solo 1 o 0 que nos indica exito o fracaso, de esta manera poder llevar la cuenta de las bolas de la sección A.

```python
def Ehrenfest(A1, A2, n):
  """
  A1 - bolas del lado izquierdo
  A2 - bolas del lado derecho
  n - numero de simulaciones
  """
  N= A1+A2
  j=A1
  estado= [j]
  for i in range(n):
    p=np.random.binomial(1, j/N)
    if p:
      j -= 1
    else:
      j += 1
    estado.append(j)
  plt.figure(figsize=(15,6))
  plt.plot(estado, marker='o',c='b')

Ehrenfest(50,40,100)
```

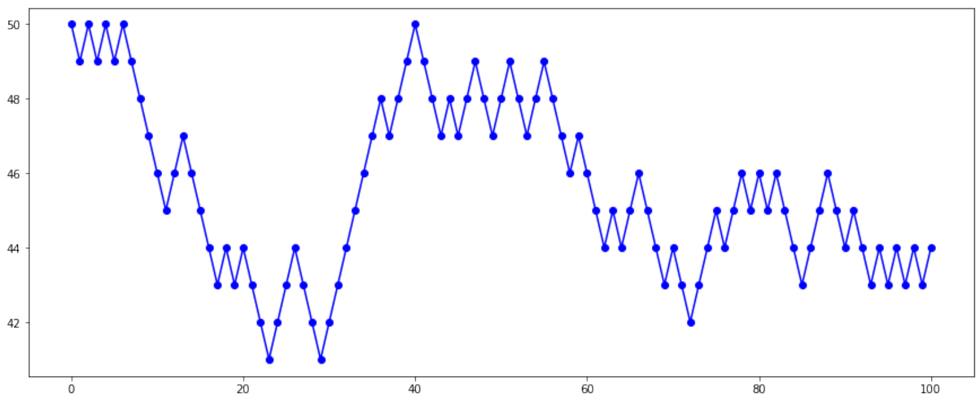

FIG. 4 – *Trayectoria para tiempo n*

Vemos que precisamente en nuestra gráfica representa la trayectoria, le hemos indicado que tenemos 50 bolas en la sección izquierda de la urna y 40 bola en la parte derecha y deseamos ver nuestro proceso para un tiempo $t = 100$,

4. Haga otro programa que simule la trayectoria del proceso hasta un tiempo de paro: Empieza inicialmente con una distribución inicial π_0, y simula la trayecto-

ria X_0,X_1,...X_T hasta un tiempo aleatorio T, definido como la primera vez que el proceso toma el estado fijo i_0 (el estado i_0 esta dado por el usuario).
En este caso utilizamos el mismo programa que el anterior, pero con una pequeña modificación ya que ahora el usuario nos debe indicar has que número de bolas en la sección izquierda de la urna debe parar nuestro programa.

```
def Ehr_paro(A1, A2, t0):
  """
  A1 - bolas del lado izquierdo de la urna
  A2 - bolas del lado derecho de la urna
  t0 - numero de bolas a parar
  """
  N= A1+A2
  j=A1
  estados= [j]

  while j != t0:
    p=np.random.binomial(1, j/N)
    if p==1:
      j -= 1
    else:
      j += 1
    estados.append(j)
  plt.figure(figsize=(15,6))
  plt.plot(estados,c='b')
  plt.show()

Ehr_paro(30,20,25)
```

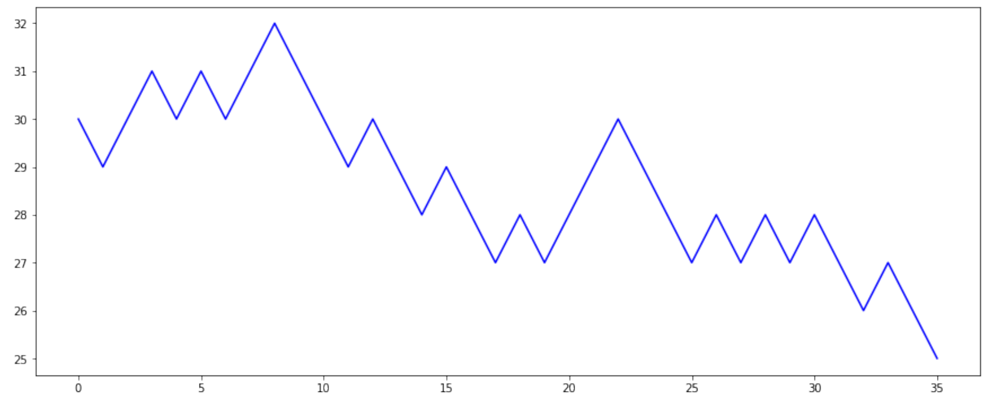

Fig. 5 – *Trayectoria para una cantidad especificas de bolas*

Vemos en este caso que le hemos indicado que tenemos en la sección izquierda de nuestra caja 30 bolas y la sección derecha 20 y deseamos parar cuando tengamos sólo 25, y vemos en nuestra gráfica que precisamente se detiene a los 25.

5. Usa el algoritmo anteriror para contestar lo siguiente para varios valores de N:

```python
def ETiempos(A1, A2, t0, K):
  N= A1+A2
  ts=[]
  for  i in range (K):
    k=A1
    cont=0
    while k != t0:
      p=np.random.binomial(1, k/N)
      if p==1:
        k -= 1
      else:
        k += 1
      cont+=1

  ts.append(cont)
  t = sum(ts)/ len(ts)
  print(f'Para {K} simulaciones el tiempo promedio' +
  + f'desde {A1} a {t0} fue de {t} milisegundos')

ETiempos(0,10,5,100)
ETiempos(0,100,50,100)
ETiempos(0,500,250,100)
ETiempos(10,10,5,100)
# ETiempos(10,10,0,100) (tardo demasiado en terminar)
# ETiempos(0,500,500,100) (tardo demasiado en terminar)
```

(a) Dado que el proceso empezó en 0, ¿Cuánto tarda en promedio en llegar al estado $\lfloor \frac{N}{2} \rfloor$?

(b) Dado que el proceso empezó en 0, ¿Cuánto tarda en promedio en llegar al estado N?

(c) Dado que el proceso empezó en N, ¿cuánto tarda en promedio en llegar al estado $\lfloor \frac{N}{2} \rfloor$?

(d) Dado que el proceso empezó en N, ¿Cuánto tarda en promedio en llegar al estado 0?

(e) ¿Qué observas en los resultados anteriores?

```
Para 100 simulaciones el tiempo promedio desde 0 a 5 fue de 7.0 ms
Para 100 simulaciones el tiempo promedio desde 0 a 50 fue de 164.0 ms
Para 100 simulaciones el tiempo promedio desde 0 a 250 fue de 618.0 ms
Para 100 simulaciones el tiempo promedio desde 10 a 5 fue de 39.0 ms
```

Vemos que conforme vamos queriendo conocer el tiempo promedio para una N va aumentando la cantidad de milisegundos, para nuestro ejemplo el tiempo de ejecución no fue tardado, pero cuando quisimos conocer el tiempo cuando un proceso empieza en N y debe llegar al estado 0 tardo el programa demasiado en contestar y no pudios obtener el tiempo, pero esto nos da una idea de que pasar a los estados 0 o N suele tardar demasiado por la entropía de las bolas, pero como vimos en clase:

$$P(\text{Exista n} \geq 1 X_n = N | X_0 = N) = 1$$

No sabremos con certeza para que n va cumplir, en este caso, no sabremos cuando terminara, pero esta probabilidad se cumple si vemos una urna de dos bolas, ya que al llegar al estado N sera más rápido debido a que sólo tenemos dos bolas y viceversa con el estado 0.

6. Fija un N y calcula, para n muy grande, la densidad de probabilidades de la variable aleatoria X_n (aproxime por un histograma y después conjeture una densidad conocida con sus parámetros).

Nota: Para las estimaciones (preguntas 5 y 6) no es necesario que se guarde toda la trayectoria.

En esta práctica pueden utilizar Python o R. No olviden poner el número de alumno en Moodle, y si desean poner su nombre que sea empezando por el apellido paterno pues así esta en la lista.

[1] L.Rincon.(2012).*Introducción a los Procesos Estocásticos.* Las prensas de ciencias, México.