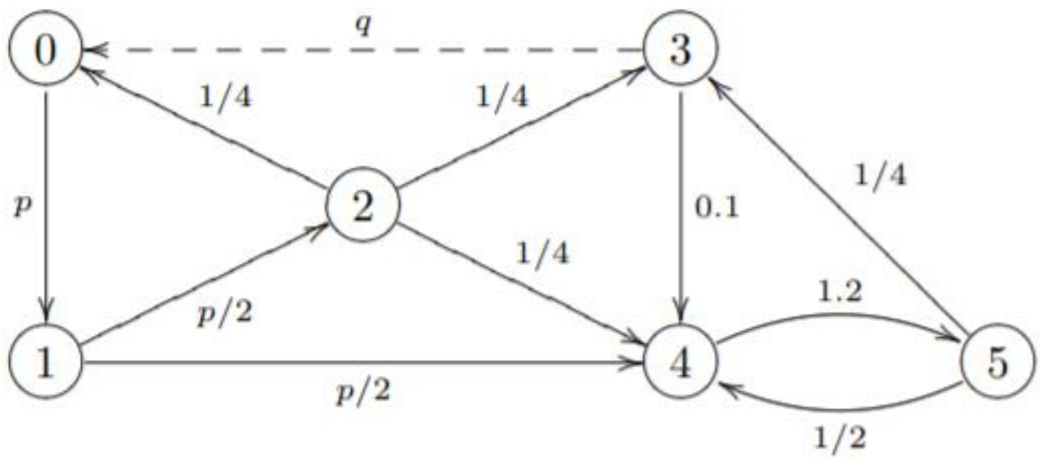
ADRIÁN RUBIO PINTADO

18/11/2021

Stochastic Systems --- Discrete Time Systems

Ejercicios --- Prueba intermedia

Consideremos la siguiente cadena de Markov:



(se asuma que en cada estado hay una flecha hasta el mismo estado que hace que la suma de las probabilidades en salida sea 1), con p = 0.3.

Nota: las medidas son medias de conjunto de conjunto de conjunto, no medidas en el tiempo. Esto quiere decir que no se puede crear una cadena y hacer medidas en el tiempo según evolucionan los estados: hay que crear un conjunto de cadenas, ejecutarlas en paralelo hasta un instante t y luego hacer las medidas sobre el conjunto de estados en que las cadenas han llegado.

i) Determinar la matriz de transición de esta cadena.

P.Estado->Estado	Estado 0	Estado 1	Estado 2	Estado 3	Estado 4	Estado 5
Estado 0	0.7	0.3	0	0	0	0
Estado 1	0	0.7	0.15	0	0.15	0
Estado 2	0.25	0	0.25	0.25	0.25	0
Estado 3	**q**	0	0	**0.9-q**	0.1	0
Estado 4	0	0	0	0	0.5	0.5
Estado 5	0	0	0	0.25	0.5	0.25

Dado p=0.3 , cada celda determina la probabilidad de pasar de un estado a otro, es decir P(m->n)

ii) Simular el funcionamiento de la cadena y hacer una estimación de conjunto de h 2 0 y h 5 0 para q = 0.1 y q = 0.

**Para estas simulaciones se utilizará el código ya implementado para simular cadenas de Markov que se puede consultar en [1].*

Se define

$$h_m^A = P(H^A < \infty)$$

es decir, la probabilidad de que partiendo del estado m, lleguemos a pasar por el conjunto de estados A alguna vez(probabilidad < ∞). Por ello simulamos varias cadenas hasta un tiempo suficientemente grande t, y hacemos un promedio de conjunto para obtener la probabilidad de pasar al menos una vez por el estado deseado.

Simulamos y obtenemos los siguientes resultados:*(Simulamos con un tiempo máximo de <u>t = 10^3</u> con n = 10^3 cadenas)*

	h_0^2	h_0^5
q = 0.1	1	1
q = 0	0.498	1

iii) Simular el funcionamiento de la cadena y hacer una estimación de conjunto de k 2 0 y k 2 4 para q = 0.1 y q = 0.

**Para estas simulaciones se utilizará el código ya implementado para simular cadenas de Markov que se puede consultar en [1].*

Simulamos y obtenemos los siguientes resultados:*(Simulamos con un tiempo máximo de <u>t = 10^3</u> con n = 10^3 cadenas)*

	k_0^2	k_0^4
q = 0.1	42	71
q = 0	522	1000

iv) Usar el sistema de ecuaciones lineares oportuno para determinar los valores teóricos correspondientes a las cantidades estimadas y comparar con los valores determinados por medio de la simulación (cuidado: si una cantidad k es ∞ la simulación claramente no puede dar su valor real... discutir este caso).

El vector de probabilidades de paso(o “hitting probabilities”) a calcular de:

$$A \subseteq M,\ h^A = [h_m^A | m \in M]'$$

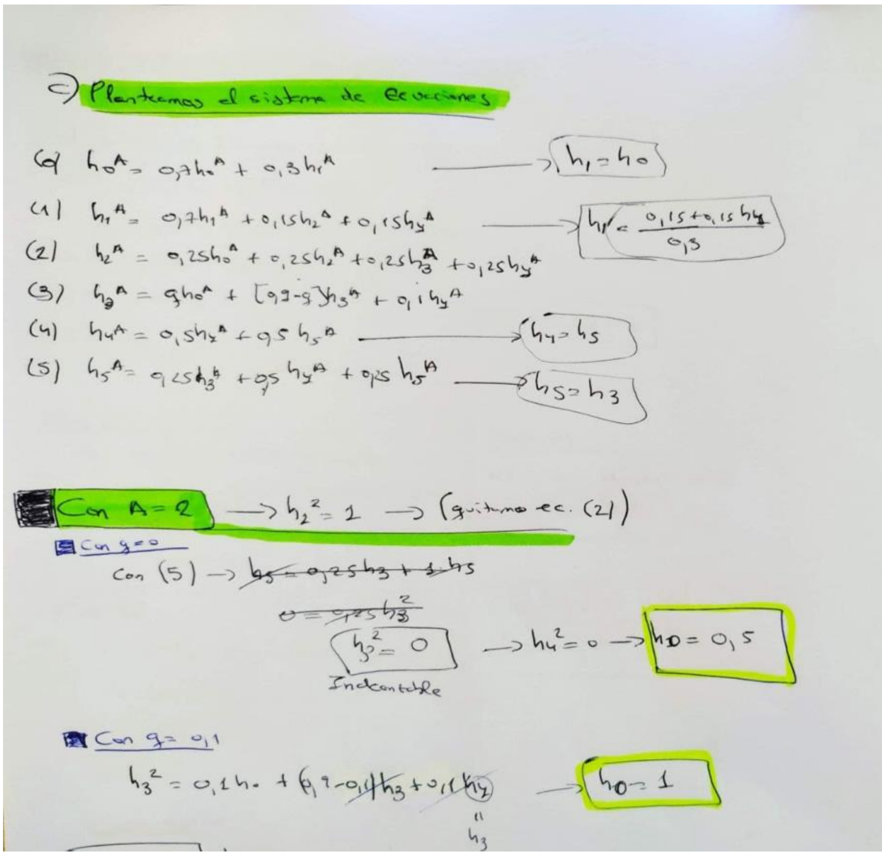
es la solución mínima no negativa de:

$$\begin{aligned} h_m^A &= 1 & m \in A \\ h_m^A &= \sum_{n \in M} \mathbf{P}_{m,n} h_n^A & m \notin A \end{aligned}$$

Obtenemos los siguientes resultados teóricos:

	h_0^2	h_0^5
q = 0.1	1	1
q = 0	0.5	1

Para obtener dichos resultados, hemos planteado los siguientes sistemas de ecuaciones:



Planteamos el sistema de ecuaciones para las K. (Notación: $K_x = K_x^A$)

PARA A = 2 $\longrightarrow \boxed{K_2 = 0}$

(0) $K_0 = 1 + 0,7K_0 + 0,3K_1 \longrightarrow \boxed{K_0 = \frac{1}{0,3} + K_1}$

(1) $K_1 = 1 + 0,7K_1 + \cancel{0,15K_2} + 0,15K_4$

(2) $K_2 = 0$

(3) $K_3 = 1 + gK_0 + [0,9 - g]K_3 + 0,1K_4$

(4) $K_4 = 1 + 0,5K_4 + 0,5K_5 \longrightarrow \boxed{K_4 = 2 + K_5}$

(5) $K_5 = 1 + 0,25K_3 + 0,5K_4 + 0,25K_5$

Con (5) y (4)

$K_5 = 1 + 0,25K_3 + 1 + 0,5K_5 + 0,25K_5$

...

$\boxed{K_5 = 8 + K_3}$ (7)

Con (1)

$0,3K_1 = 1 + 0,15K_4$

...

$\boxed{K_1 = \frac{22}{3} + \frac{K_3}{2}}$ (8)

Con (3)

$K_3 = 1 + gK_0 + K_3 \cdot 0,9 - gK_3 + 0,1K_4$

...

$\boxed{K_3 = K_0 + \frac{9}{5g}}$ (9)

Con (8), (9), (0)

$K_1 = \frac{22}{3} + \frac{K_0 + \frac{9}{5g}}{2}$

...

$\boxed{K_0 = 16 + \frac{27}{20g}}$

Con g = 0

$K_0 = 16 + \frac{27}{20 \cdot 0} = \infty$ $\boxed{K_0^2 = \infty}$

$\boxed{K_0 = \infty} \longrightarrow K_3 = \infty \longrightarrow K_5 \rightarrow \infty \longrightarrow \boxed{K_4^2 = \infty}$

Con g = 0,1

$K_0 = 16 + \frac{27}{20 \cdot 0,1} = 29,5$ $\boxed{K_0^2 = 29,5}$

$K_3 = 29,5 + \frac{9}{5 \cdot 0,1} = 47,5$

$K_5 = 8 + 47,5 = 55,5$

$K_4 = 2 + 55,5 = 57,5$ $\boxed{K_4^2 = 57,5}$

De manera análoga, plantemos los sistemas de ecuaciones para los vectores “hitting times” medios:

$$f\ A \subseteq M,\ k^A = [k_m^A | m \in M]'$$

Tal que:

$$\begin{array}{lll} k_m^A & = 0 & m \in A \\ k_m^A & = 1 + \sum_{n \in M} \mathbf{P}_{m,n} k_n^A & m \notin A \end{array}$$

Obtenemos los siguientes resultados teóricos:

	k_0^2	k_4^2
q = 0.1	29.5	57.5
q = 0	∞	∞

Observamos que para los casos q = 0 el resultado teórico es infinito. Sin embargo, en la simulación hemos simulado hasta un tiempo máximo t = 1000 . En el caso k_4^2 obtenemos un resultado de t=1000 de simulación obtenido al promediar sobre todas las cadenas con t=1000, es decir, en todos los casos el tiempo es el máximo. Si simuláramos con un tiempo cada vez más grande, la salida de la simulación tendrá el valor de t máximo pasado en la simulación. Es decir, nunca converge y equivale al tiempo teórico(infinito.)

Para el caso de $\mathbf{k_0^2}$ también obtenemos un valor teórico de infinito. En la simulación obtenemos un resultado de 500 aprox, es decir, t_simulación/2(Teóricamente infinito/2 = infinito). Esto es debido a que al promediar, aproximadamente la mitad nunca llega a “llegar” al estado.

Para obtener los resultados teóricos, hemos planteado los siguientes sistemas de ecuaciones:

Planteamos el sistema de ecuaciones para las K. (Notación: $K_x = K_x^A$)

PARA A=2 $\longrightarrow \boxed{K_2 = 0}$

(0) $K_0 = 1 + 0{,}7K_0 + 0{,}3K_1 \longrightarrow \boxed{K_0 = \frac{1}{0{,}3} + K_1}$

(1) $K_1 = 1 + 0{,}7K_1 + \cancel{0{,}15K_2} + 0{,}15K_4$

(2) $K_2 = 0$

(3) $K_3 = 1 + qK_0 + [0{,}9 - q]K_3 + 0{,}1K_4$

(4) $K_4 = 1 + 0{,}5K_4 + 0{,}5K_5 \longrightarrow \boxed{K_4 = 2 + K_5}$

(5) $K_5 = 1 + 0{,}25K_3 + 0{,}5K_4 + 0{,}25K_5$

Con (5) y (4)

$K_5 = 1 + 0{,}25K_3 + 1 + 0{,}5K_5 + 0{,}25K_5$

...

$\boxed{K_5 = 8 + K_3}$ (7)

Con (1)

$0{,}3K_1 = 1 + 0{,}15K_4$

...

$\boxed{K_1 = \frac{22}{3} + \frac{K_3}{2}}$ (8)

- Con (3)

$K_3 = 1 + qK_0 + K_3 \cdot 0{,}9 - qK_3 + 0{,}1K_4$

...

$\boxed{K_3 = K_0 + \frac{9}{5q}}$ (9)

- Con (8), (9), (0)

$K_1 = \frac{22}{3} + \frac{K_0 + \frac{9}{5q}}{2}$

...

$\boxed{K_0 = 16 + \frac{27}{20q}}$

Con q = 0

$K_0 = 16 + \frac{27}{20 \cdot 0} = \infty \quad \boxed{K_0^2 = \infty}$

$\boxed{K_0 = \infty} \longrightarrow K_3 = \infty \longrightarrow K_5 \to \infty \longrightarrow \boxed{K_4^2 = \infty}$

Con q = 0,1

$K_0 = 16 + \frac{27}{20 \cdot 0{,}1} = 29{,}5 \quad \boxed{K_0^2 = 29{,}5}$

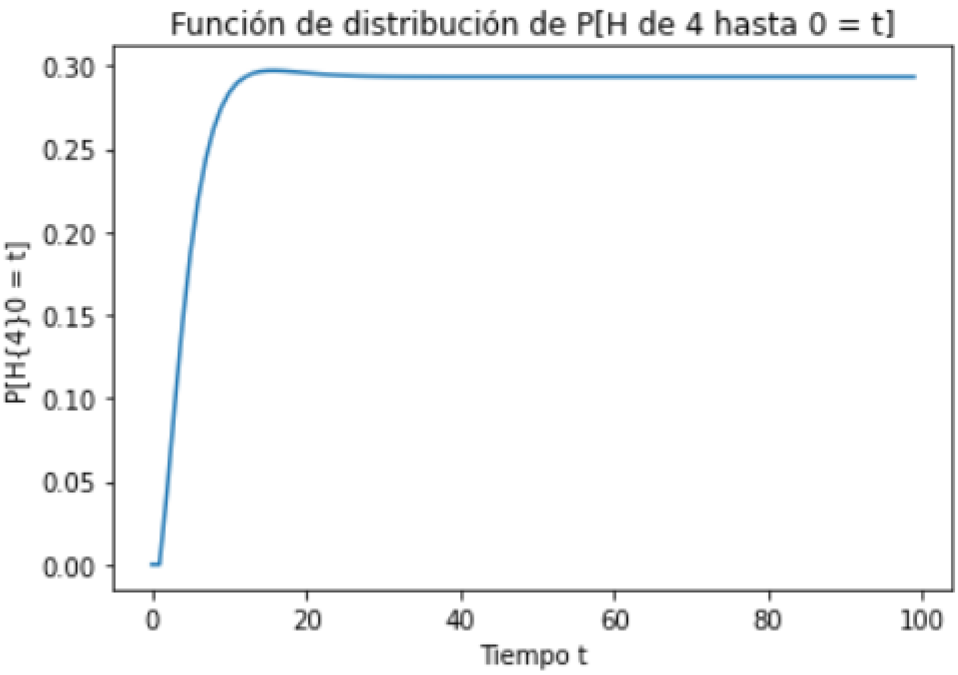
$K_3 = 29{,}5 + \frac{9}{5 \cdot 0{,}1} = 47{,}5$

$K_5 = 8 + 47{,}5 = 55{,}5$

$K_4 = 2 + 55{,}5 = 57{,}5 \quad \boxed{K_4^2 = 57{,}5}$

v) para el caso q = 0.1, dibujar el grafico de g(t) = P[H {4} 0 = t]

Dibujamos la función de distribución de la probabilidad de alcanzar 4 por primera vez en tiempo t partiendo desde el estado 0. Para ello elevamos La matriz de probabilidades al tiempo t que deseamos calcular.



En tiempo t= 0 estamos en el estado 0, ya que es desde el estado que partimos, por ello la probabilidad de alcanzar 4 por primera vez en tiempo igual a 0 es 0. Observando el gráfico, vemos como desde t=0 vemos un pequeño pie, que nos indica que hasta que no alcanzamos un tiempo 2, no existe probabilidad de alcanzar 4 (se puede observar viendo el grafo.) A partir de ahí, la probabilidad de alcanzar 4 en un tiempo menor o igual a t crece hasta estabilizarse en un 30%. Es decir decir, la probabilidad de alcanzar 4 en u tiempo menor o igual a 20 es como mucho un 30%, dada la naturaleza de la cadena.

vi) Afirmo que H {4} 0 < H{5} 0 siempre. ¿Es cierto? ¿Por qué no? ¿Por qué sí? (Elegid la opción apropiada)

Sí, ya que observando la cadena vemos como el único modo de llegar al estado 5, es pasando antes por el estado 4. Dado que el estado 4 tiene un 50% de probabilidad de volver así mismo una vez a él, la probabilidad de alcanzar el estado 4 antes del 5 es mayor. Es decir, desde el 4 puedo seguir en el 4 con 50% de probabilidad y dado un t mayor llegar al estado 5.

CÓDIGO

Utilizando el código de simulación de cadenas de markov(apartado de refencias), se ha desarrollado de manera adicional las siguientes funciones para poder obtener las medidas deseadas.

```
    def simulate_absorption_probability(self, current_state, objective_state, t=100,n=100):
        """
        Calculates absorption_probability(and mean time to reach the state) simulating n strings
            with t jumps in each string

        Parameters
        ----------
        current_state: str
            The state of the current random variable.

        objective_state: str
            The state wanted to be reached
        t: int
            The number of future states to generate.
        n: int
            The number of simulations  to generate.
        """

        n_times_reached=0
        reached_at_time = []

        for j in range(n):
            simul_current_state = current_state
            flag_time = False
            for i in range(t):#Markov string simulation
                next_state = self.next_state(simul_current_state)
                if(next_state == objective_state):
                    n_times_reached += 1
                    reached_at_time.append(i)
                    flag_time = True
                    break
                else:

                    simul_current_state = next_state
```

```
                    simul_current_state = next_state
            if(not flag_time):
              reached_at_time.append(t)


        k = int(np.mean(reached_at_time)) if len(reached_at_time)>0 else 0
        return (n_times_reached/n), k


#definimos matriz de transicion
def get_transition_prob(q):
    transition_prob =   { '0' : {'0':0.7 ,  '1': 0.3 ,  '2': 0,     '3':0 ,         '4': 0,     '5':0   },
                         '1' : {'0':0,     '1': 0.7,   '2':0.15 ,  '3': 0,         '4':0.15 ,  '5':0   },
                         '2' : {'0':0.25,  '1':0 ,     '2':0.25 ,  '3':0.25 ,      '4':0.25 ,  '5':0   },
                         '3' : {'0':q ,    '1':0 ,     '2': 0,     '3':(0.9-q),'4': 0.1 ,  '5': 0  },
                         '4' : {'0':0 ,    '1': 0,     '2': 0,     '3':0 ,         '4': 0.5,   '5':0.5   },
                         '5' : {'0':0 ,    '1': 0,     '2':0 ,     '3':0.25 ,      '4': 0.5,   '5':0.25    },
                        }
```

Un ejemplo de llamada sería:

```
[2] #MACROS
    t = 10**3#tiempo maximo de la simulacion
    n = 10**3
    print('Tiempo de simulación de las cadenas: ', t, 'Numero de cadenas simuladas:' ,n)
```

```
Tiempo de simulación de las cadenas:  1000 Numero de cadenas simuladas: 1000
```

```
#h_0_2
q = 0.1
ej1_1_chain = MarkovChain(transition_prob=get_transition_prob(q))
h_0_2 , k_0_2 = ej1_1_chain.simulate_absorption_probability( current_state = '0', objective_state='2', t=t,n=n)
print('\n Estado objetivo: ', 2, 'q = ', q)
print('\t--> Resultados simulación h_0_2: ',h_0_2,' k_0_2:   ', k_0_2 )
```

```
 Estado objetivo:  2 q =  0.1
        --> Resultados simulación h_0_2:  1.0  k_0_2:    43
```

Referencias

[1] https://medium.com/@__amol__/markov-chains-with-python-1109663f3678