

Compléments sur les chaînes de Markov

I. Chaînes de Markov et Python

Soit (X_n) une chaîne de Markov à n états et $A = (p_{i,j})_{(i,j) \in \llbracket 1, n \rrbracket^2}$ sa matrice de transition, c'est à dire :

$$\forall (i, j) \in \llbracket 1, n \rrbracket^2, p_{i,j} = \mathbb{P}_{(X_n=i)}(X_{n+1} = j)$$

On écrit alors un programme qui simule la trajectoire des k premiers pas de la chaîne (k est à choisir par l'utilisateur) avec la position initiale x_0 (à choisir aussi par l'utilisateur).

```
import numpy as np

def markov(k,A,x0):
    X=np.zeros(k+1)
    X[0]=x0
    E=[ i for i in range(1,len(A)+1)]
    for i in range(k):
        X[i+1]=np.random.choice(E,p=A[int(X[i])-1],:)]
    return(X)
```

Retenir

La fonction `np.random.choice(E, p=...)` simule une variable aléatoire avec support dans E (liste des états) et dont les probabilités sont contenues dans p . Attention à bien faire correspondre le E et avec la taille de la matrice M .

Dans l'exemple précédent,

- On simule une variable à support dans $\llbracket 1, n \rrbracket$ (n est la taille de la matrice de transition). Sur Python l'ensemble des états de la chaîne à $\llbracket 0, n-1 \rrbracket$. On rappelle que si A est une matrice carrée de taille $n \times n$, alors la commande `len(A)` renvoie l'entier n .
- Dans le i -ième tour de boucle, la chaîne se trouve donc dans l'état $X[i]$ et la loi de $X[i+1]$ s'obtient donc en conditionnant par rapport à $X[i]$. Les valeurs de la loi de X_{i+1} conditionnée à $X_k = X[k]$ sont sur la ligne $X[i]$ de la matrice, que l'on extrait donc avec la commande `A[int(X[i])-1],:]`.

II. Déplacement d'un mobile

Un mobile se déplace entre les quatre sommets d'un carré numérotés 1, 2, 3 et 4 (dans le sens horaire). On note X_n la variable aléatoire donnant le numéro du sommet sur lequel se trouve le mobile à l'étape n et on note

$$U_n = (P(X_n = 1) \quad P(X_n = 2) \quad P(X_n = 3) \quad P(X_n = 4))$$

À chaque étape la probabilité de passer d'un sommet a à un sommet adjacent est de $\frac{1}{4}$ et celle de passer à un sommet opposé est de $\frac{1}{2}$. À l'instant initial $n = 0$ le mobile se trouve en position 2

1. Représenter la situation à l'aide d'un graphe. Les arêtes seront pondérées avec les probabilités de transition.
2. Déterminer une matrice M telle que $U_{n+1} = U_n M$. En déduire que pour tout $n \in \mathbb{N}$ $U_n = U_0 M^n$.
3. Déterminer les états stables de cette chaîne de Markov.
4. Que fait la fonction suivante ?

```
M=np.array([[0,1/4,1/2,1/4],[1/4,0,1/4,1/2],[1/2,1/4,0,1/4],[1/4,1/2,1/4,0]])

def markov(M,x0,n):
    E=[ i for i in range(1,len(M)+1)]
    position=x0
    R=[x0]
    for i in range(1,n+1):
        position=rd.choice(E,p=M[position-1])
        R.append(position)
    return R

print(markov(M,2,10))
```

5. On aimerait savoir où se trouve "le mobile en moyenne". Recopier et compléter la fonction suivantes afin qu'elle renvoie la fréquence d'apparition d'une position x dans une liste P qui contient les positions successives.

```
def freq(P,x):
    c=...
    for .....:
        if x==.....:
            c=.....;
    return .....
```

6. Utiliser la fonction précédente pour écrire un programme permettant de calculer la fréquence de chacune des positions lors de 5000 simulations. On obtient le résultat suivant :

```
[0.2443511297740452, 0.24935012997400519, 0.25574885022995403,
0.2505498900219956]
```

Comment interpréter ce résultat ?

III. Evolution d'une maladie

On modélise l'évolution d'une maladie au sein d'une population au cours du temps en classant les individus en trois groupes :

- Le groupe S des individus sains et non immunisés ;
- Le groupe M des individus malades ;
- Le groupe I des individus immunisés.

On appelle $E = \{S, M, I\}$ l'ensemble des états possibles. On donne la loi d'évolution suivante :

- À l'instant 0, tous les individus sont dans l'état S .
- La moitié des individus dans l'état S à l'instant n restent dans le même état à l'instant $n + 1$ et l'autre moitié tombe malade ; un dixième des individus immunisés à l'instant n passent dans le groupe S à l'instant $n + 1$, les autres restent immunisés. Un malade sur cinq le reste, les autres guérissent et deviennent immunisés.

1. Représenter le modèle précédemment décrit par un graphe ;
2. On s'intéresse à la suite (X_n) de variables aléatoires telle que X_n désigne l'état du système l'instant n , c'est à dire que $X_n(\Omega) = \{1, 2, 3\}$, où $(X_n = 1)$ (resp. 2,3) correspond à l'état S (resp. M, I) à l'instant n . La suite (X_n) est donc une chaîne de Markov.
 - (a) Recopier et compléter le programme ci-dessous pour qu'il simule la suite des états $X_0, X_1, \dots, X_n$ (c'est-à-dire une trajectoire de la chaîne).

```
def Exemple(n):
    X=np.zeros(n)
    X[0]=1
    for k in range(n-1):
        if X[k]==1:
            X[k+1]=np.random.choice(np.array([1,2,3]),...)
        elif X[k]==2:
            X[k+1]=...
        else:
            X[k+1]=...
    return(X)
```

- (b) Que fait le programme suivant ?

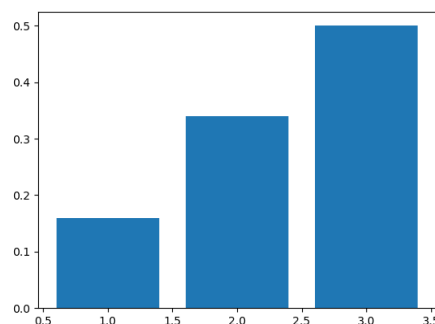
```
y=Exemple(1000)
plt.plot(y, '+')
plt.show()
```

- (c) Que fait le programme suivant ?

```
U=np.zeros(100)
for k in range(100):
    U[k]=Exemple(1000)[999]

frequ=np.zeros(3)
for j in range(3):
    for k in range(len(U)):
        if U[k]==j+1:
            frequ[j]=frequ[j]+1
frequ=frequ/len(U)
plt.bar([1,2,3], frequ)
plt.show()
```

Interpréter l'affichage qu'il donne :



3. Pour $n \in \mathbb{N}$, on note

$$U_n = (P(X_n = 1), P(X_n = 2), P(X_n = 3)).$$

- (a) Déterminer la matrice de transition de la chaîne de Markov (elle vérifie donc $U_{n+1} = U_n A$).
- (b) Montrer que, pour tout $n \in \mathbb{N}$, on a $U_n = U_0 A^n$. Préciser U_0 , (c'est la loi initiale).
- (c) Ecrire un programme Python donnant la matrice de transition A et le vecteur de la loi initiale u_0 , et qui calcule $U_0 A^{1000}$.

IV. Algorithmme du Page Rank de Google.

Nous allons illustrer la méthode utilisée par Google pour classer les pages web par "indice de popularité". L'idée est d'étudier l'évolution des déplacements d'un individu sur internet, et de regarder la probabilité p_i d'être sur un site i au bout d'un très grand nombre de déplacements. On dit alors que p_i est alors appelée "indice de popularité de la page i ". Nous commencerons par regarder un cas très simple où il n'existerait sur le Web que 3 pages internet puis on étudiera le cas général.

Un exemple avec 3 sites. À l'issue de sa requête, un internaute est susceptible d'aller sur trois sites A , B , et C . On suppose que l'internaute se déplace forcément vers un lien de la page Web où il se trouve.

- le site A contient un lien vers lui-même, un lien vers B , et un lien vers C ;
- le site B contient 5 liens vers lui-même, un vers A et un vers C ;
- le site C comporte un lien vers A , 7 liens vers B et 4 vers C .

À l'instant 0, l'internaute choisit au hasard l'un des trois sites. Par la suite, la probabilité de passer d'un site (à l'instant n) vers un autre (à l'instant $n+1$) est proportionnelle au nombre de liens du premier site vers le deuxième.

Pour $n \in \mathbb{N}$, on désigne par a_n, b_n et c_n les probabilités des événements A_n (resp. B_n, C_n) définis respectivement par le fait de se trouver sur le site A (resp. B, C) à l'instant n .

Enfin, on note X_n la variable aléatoire définie par

$$X_n(\omega) = \begin{cases} 1, & \text{si } \omega \in A_n \\ 2, & \text{si } \omega \in B_n \\ 3, & \text{si } \omega \in C_n \end{cases}$$

1. En justifiant rigoureusement au moins l'une des égalités, exprimer a_{n+1} , (resp. b_{n+1}, c_{n+1}) en fonction des trois réels a_n, b_n et c_n .
2. Écrire la matrice de transition S .
3. Compléter le programme suivant qui définit S et permet d'afficher la trajectoire $(X_0, X_1, \dots, X_n)$ de la chaîne de Markov (X_n) .

```
S= ...
```

```
def navigation(n,S):
    X=np.zeros(n)
    X[0]=...
    for k in range(n-1):
        X[k+1]=np.random.choice(a=list(range(1,4)),p=...)
    return(X)
```

4. Compléter le programme suivant afin de définir une matrice E de taille 3×11 dont la i -ème colonne contient le vecteur $i-1$ -ème état probabiliste.

```
E=np.zeros([3,11])
E[:,0]=(1/3)*np.ones([3])
for k in range(10):
    E[:,k+1]=...
```

5. Écrire un programme qui trace sur un même graphique les courbes des suites (a_n) , (b_n) et (c_n) . Pour $n \in \llbracket 0, 10 \rrbracket$. On obtient :
6. On appelle "indice de notoriété" d'une page A la valeur $a = \lim a_n$. Déterminer la valeur exacte des indices de notoriété en calculant la valeur exacte de l'état stable. On admettra que la probabilité invariante est unique.