

Corrigé

Code de partage avec Capytale : 17a2-9457404

On utilisera les bibliothèques suivantes :

```
import numpy as np
import numpy.linalg as al
import numpy.random as rd
import matplotlib.pyplot as plt
```

Illustration sur un exemple

Doudou le hamster passe son temps entre ses trois activités favorites : dormir, manger et faire du sport dans sa roue. Au début de la journée, il mange, et à chaque heure, il change d'activité selon les critères suivants.

- si, à l'heure n , il est en train de manger, alors il va dormir l'heure suivante avec probabilité 0,7 et faire de l'exercice avec probabilité 0,3
- si, à l'heure n , il est en train de dormir, alors il continue à dormir l'heure $n + 1$ avec probabilité 0,4, il va manger avec probabilité 0,3 et il va faire de l'exercice avec probabilité 0,3
- si, à l'heure n , il est en train de faire de la roue, il va manger l'heure suivante avec probabilité 0,5 et il va dormir avec probabilité 0,5

On s'intéresse ici à l'évolution du comportement de Doudou. On souhaite notamment déterminer si l'une de ses activités prendra, à terme, le dessus sur les autres.

Modélisation mathématique

On modélise ce problème comme suit :

- on commence par numérotiser les activités par un entier entre 1 et 3
- on note X_n la v.a.r. égale à l'état du hamster à l'heure n

Ainsi, la suite de v.a.r. (X_n) représente l'évolution des activités du hamster.

- Cette évolution peut être modélisée par un graphe, que l'on représentera.
- On définit enfin la matrice de transition A associée au problème. Il s'agit de la matrice :

$$A = (a_{i,j}) \text{ où } a_{i,j} = P_{[X_n=i]}(X_{n+1} = j)$$

($a_{i,j}$ représente la probabilité de passage de l'état i à l'état j)

1 Etude de la matrice de transition

1. Déterminer $P_{[X_0=i]}(X_1 = j)$

A quel coefficient de la matrice A cela correspond-il ?

On note 1, 2 et 3 les activités : manger, dormir et faire du sport. Alors, d'après les données de l'énoncé :

$$\begin{array}{lll} P_{[X_0=1]}(X_1 = 1) = 0 & P_{[X_0=1]}(X_1 = 2) = 0,7 & P_{[X_0=1]}(X_1 = 3) = 0,3 \\ P_{[X_0=1]}(X_1 = 1) = 0,3 & P_{[X_0=1]}(X_1 = 2) = 0,4 & P_{[X_0=1]}(X_1 = 3) = 0,3 \\ P_{[X_0=1]}(X_1 = 1) = 0,5 & P_{[X_0=1]}(X_1 = 2) = 0,5 & P_{[X_0=1]}(X_1 = 3) = 0 \end{array}$$

2. Déterminer la matrice de transition du problème.

Ecrire, avec Python, la commande permettant de la définir et la stocker dans une variable **A**

De fait la matrice vaut $A = \begin{pmatrix} 0 & 0,7 & 0,3 \\ 0,3 & 0,4 & 0,3 \\ 0,5 & 0,5 & 0 \end{pmatrix}$ que l'on traduit avec Python par :

```
A=np.array([[0,0.7,0.3],[0.3,0.4,0.3],[0.5,0.5,0]])
```

2 Etude de l'évolution du comportement du hamster

Dans la suite, on considère la matrice ligne U_n qui définit la loi de X_n :

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

3. Déterminer la probabilité $P(X_{n+1} = 1)$ en fonction de $P(X_n = 1)$, $P(X_n = 2)$, $P(X_n = 3)$ et des coefficients de la matrice A

4. Déterminer de même $P(X_{n+1} = 2)$ et $P(X_{n+1} = 3)$

En déduire que pour tout $n \in \mathbb{N}$, $U_{n+1} = U_n A$

Exprimer enfin U_n en fonction de A et de U_0

$(X_n = 1), (X_n = 2), (X_n = 3)$ est un système complet d'événements, donc d'après la formule des probabilités totales

$$P(X_{n+1} = 1) = P(X_n = 1)P_{[X_n=1]}(X_{n+1} = 1) + P(X_n = 2)P_{[X_n=2]}(X_{n+1} = 1) + P(X_n = 3)P_{[X_n=3]}(X_{n+1} = 1) = 0,3P(X_n = 2) + 0,5P(X_n = 3)$$

$$\text{et de même } P(X_{n+1} = 2) = 0,7P(X_n = 1) + 0,4P(X_n = 2) + 0,5P(X_n = 3)$$

$$\text{et } P(X_{n+1} = 3) = 0,3P(X_n = 1) + 0,3P(X_n = 2)$$

5. Que réalise l'opération $A * A$? Et l'opération $A ** 2$?

Calculer alors A^5 , A^{10} et A^{20} . Que remarque-t-on ?

Les opérations $A * A$ et $A ** 2$ ne renvoient pas A^2 mais le produit des coefficients deux à deux :

$$a_{1,1} \times a_{1,1}, \quad a_{1,2} \times a_{1,2} \dots$$

pour obtenir A^2 , il faut utiliser `np.dot(A,A)` ou `al.matrix_power(A,2)`

donc avec `al.matrix_power(A,5)`, `al.matrix_power(A,10)`, `al.matrix_power(A,20)` on obtient A^5 , A^{10} et A^{20} et on remarque que les valeurs semblent converger, entre autres avec des valeurs proches de $\frac{1}{2}$ sur la deuxième colonne.

6. Ecrire une fonction `simuMarkovEtape(x0,A)` qui prend en paramètre l'état initial **x0** et la matrice de transition **A** et renvoie une simulation de la v.a.r. X_1

Pour chaque ligne, on utilise `rd.random()`, et la fonction renvoie 1, 2 ou 3 relativement aux probabilités définies dans l'énoncé. On définit une variable i qui vaut $x_0 - 1$ car si x_0 vaut 1 par exemple, il faut utiliser la ligne 1 :

```
def simuMarkovEtape(x0,A):
    i=x0-1 # indice de ligne
    r=rd.random()
    if r<A[i,0]:
        return 1
    elif A[i,0]<=r<A[i,0]+A[i,1]:
        return 2
    else :
        return 3
```

7. Evaluer `simuMarkovEtape(2,A)` une dizaine de fois de suite.

Le résultat obtenu paraît-il cohérent ?

On peut utiliser la commande synthétique pour créer une liste de 10 réalisations :

```
[simuMarkovEtape(2,A) for i in range(10)]
```

auquel on peut éventuellement appliquer `np.mean(...)` et on trouve une moyenne proche de 2 (encore plus si on augmente le nombre de réalisations : 100, 1 000, ...).

On obtient plus souvent le résultat 2 ce qui est logique car c'est la position qui recueille le plus de probabilités (dormir est l'activité la plus fréquente du hamster).

8. Ecrire une fonction `simuMarkov(x0,A,n)` qui prend en paramètre l'état initial `x0` et la matrice de transition `A` et renvoie une simulation de la v.a.r. X_n

La fonction précédente permet de simuler en fait une étape entre un instant n et un instant $n+1$, puisqu'il s'agit d'une chaîne de Markov homogène (la transition ne dépend pas de n). Il suffit donc de répéter la transition autant que fois que souhaité pour obtenir une simulation de X_n en fonction la situation initiale $x0$

```
def simuMarkov(x0,A,n):  
    for i in range(1,n+1):  
        x0=simuMarkovEtape(x0,A)  
    return x0
```

3 Comportement asymptotique du hamster

On souhaite conjecturer le comportement de la loi de X_n lorsque $n \rightarrow +\infty$

Pour ce faire, on compare :

- la valeur théorique de U_n (loi de X_n) pour n grand
- une valeur approchée de U_n obtenue en recueillant les effectifs de chaque état à la suite de N simulations de trajectoires de taille n

9. Partant de l'état initial `x0`, par quelle commande obtient-on U_n ?

Pour obtenir U_n , c'est à dire les probabilités pour la variable aléatoire à l'instant n , on démontre mathématiquement que $U_n = U_0(A^n)$, et avec Python, par exemple pour `x0=1`, on obtient ces probabilités avec la commande : `np.dot(np.array([[1,0,0]]),al.matrix_power(A,n))`

10. Compléter et exécuter le programme suivant.

```
# Valeur des paramètres  
N = 1000  
n = 50  
x0 = 2  
# Distribution théorique  
P =  
# Valeurs observées  
Obs = [... for k in range(N)]  
# Tracés des histogrammes  
cl = [0.5, 1.5, 2.5, 3.5]  
# Diagramme des fréquences observées  
plt.hist(Obs, bins = cl)
```

On pourra le compléter encore pour faire apparaître la distribution théorique

Il suffit de simuler la variable aléatoire à l'aide de la fonction de la question 8.

`Obs = [simuMarkov(x0,A,n) for k in range(N)]` Le `P` n'est pas nécessaire dans un premier temps, mais on peut l'utiliser pour représenter les probabilités théoriques (dans un second temps), grâce à la commande de la fonction précédente et avec l'utilisation de `plt.bar`

```
P = np.dot(np.array([[0,1,0]]),al.matrix_power(A,n))  
probas=[P[0,0],P[0,1],P[0,2]] # pour transformer le P en liste  
plt.bar([1,2,3],probas)  
plt.show()
```

11. Expliquer le fonctionnement de la fonction `plt.hist` du programme précédent.

La fonction `plt.hist` a deux arguments : une première liste de valeurs : celle de la série statistique étudiée et une deuxième qui crée les classes de valeurs (il s'agit d'une liste de valeurs, dont chaque couple de valeurs consécutives constitue une classe de valeurs). La fonction `plt.hist` représente ensuite le nombre de valeurs appartenant à chaque classe.

12. Quel diagramme obtient-on si l'on part initialement avec un autre état x_0 ?

On obtient le même diagramme, on ne remarque pas de différence, puisqu'après 50 étapes, le hamster sera en moyenne la moitié du temps en position 2 et approximativement un quart du temps sur chacune des deux autres.

13. Que peut-on en conclure sur le comportement à terme du hamster ?

Comme évoqué à la question précédente, à terme, le hamster aura passé la moitié du temps en position 2

14. Si l'on est capable de démontrer que la suite (U_n) converge vers une limite notée U , quelle relation peut-on établir entre U et A ?

Il s'agit alors d'un état stable, qui vérifie $U = UA$