



## Travaux Pratiques N°2

Méthodes numériques et statistiques: Simulation d'une chaîne de Markov  
Anicet E. T. Ebou

---

### 1. Rappels

Une chaîne de Markov est un processus aléatoire  $(X_n)_{n \in \mathbb{N}}$  dont les transitions sont données par une matrice stochastique  $P(X_n, X_{n+1})$ . Ces processus vérifient la propriété de Markov, c'est-à-dire qu'observés à partir d'un temps (d'arrêt)  $T$ ,  $(X_{T+n})_{n \in \mathbb{N}}$  ne dépend que de  $X_T$  et est de nouveau une chaîne de Markov.

De façon plus algorithmique une chaîne de Markov est classiquement définie comme une suite de variables aléatoires pour laquelle la meilleure prédiction que l'on puisse faire pour l'étape  $n + 1$  si on connaît toutes les valeurs antérieures est la même que si on ne connaît que la valeur à l'étape  $n$  (le futur et le passé sont indépendants conditionnellement au présent). Nous partons ici d'une définition moins classique, mais plus proche des applications.

### 2. 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:

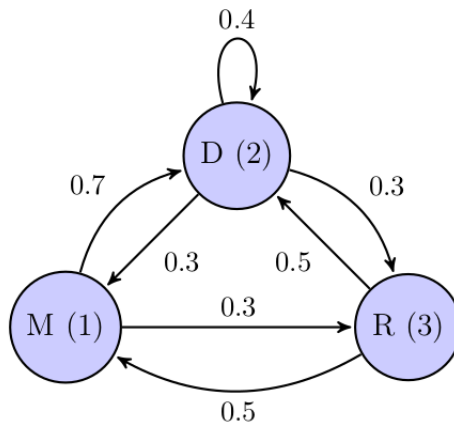
1. Si, à l'heure  $n$ , il est en train de manger, alors il va dormir l'heure suivante avec probabilité 0.7 et faire de l'exercice avec probabilité 0.3.
2. Si, à l'heure  $n$ , il est en train de dormir, alors il continue à dormir l'heure  $n+1$  avec probabilité 0.4, il va manger avec probabilité 0.3 et il va faire de l'exercice avec probabilité 0.3.
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.

### 3. Modélisation mathématique

On modélise ce problème comme suit.

- On commence par numéroter 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 le graphe suivant.



- On définit enfin la matrice de transition  $A$  associée au problème. Il s'agit de la matrice

$$A = (a_{i,j}) \text{ où } a_{i,j} = \mathbb{P}(X_n = j | X_{n+1} = i)$$

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

#### 4. Etude de la matrice de transition et de l'évolution du comportement du hamster

Avant d'entamer l'étude en **Python** de ce problème, on importe les modules nécessaires.

```
import numpy as np
```

##### Exercice 1

Déterminer la matrice de transition du problème précédent et écrire l'appel permettant de la stocker dans une variable  $A$ .

##### Exercice 2

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

##### Exercice 3

Déterminer de même  $\mathbb{P}(X_{n+1} = 2)$  et  $\mathbb{P}(X_{n+1} = 3)$ . En déduire que pour tout  $n \in \mathbb{N}$ ,  $U_{n+1} = A \times U_n$ . Exprimer enfin  $U_n$  en fonction de  $A$  et de  $U_0$ .

##### Exercice 4

Que réalise l'opération  $A * A$  si  $A$  est de type `array`?

Calculer  $A^5$ ,  $A^{10}$  et  $A^{20}$  à l'aide de la fonction `np.linalg.matrix_power`.

Que remarque-t-on?

##### Exercice 5

Écrire une fonction `simulation_markov_etape(x0,A)` qui prend en paramètre l'état initial  $x_0$  et la matrice de transition  $A$  et renvoie une simulation de la v.a.r.  $X_1$ .

Évaluer `simulation_markov_etape(2,A)` une dizaine de fois de suite. Le résultat obtenu paraît-il cohérent ?

### Exercice 6

Écrire une fonction `simulation_markov(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.  $X_n$ .

## 5. 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  $n$ .

### Exercice 7

Partant de l'état initial  $x_0$ , par quel appel obtient-on  $U_n$ ?

```
import matplotlib.pyplot as plt
# Valeur des paramètres
N = 1000
n = 50
x0 = 2

# Distribution théorique
P = ...

# Valeurs observées
Obs = [... for k in range(N)]

# Tableau des effectifs observés
cl = np.linspace(1,3,3)
effectif = calcEffectif(cl, Obs)

# Tracés des diagrammes
absc = np.linspace(1,3,3)
# Diagramme de la distribution théorique
plt.bar(absc, P, color = 'r', width = 0.2)
# Diagramme des fréquences observées
plt.bar(absc+0.2, effectif/N, color = 'b', width = 0.2)
plt.show()
```

### Exercice 8

Quel diagramme obtient-on si l'on part initialement avec un autre état `x0`?  
Que peut-on en conclure sur le comportement à terme du hamster ?